La Commode

Le Magazine des Ordinateurs Commodore Atari et Oric

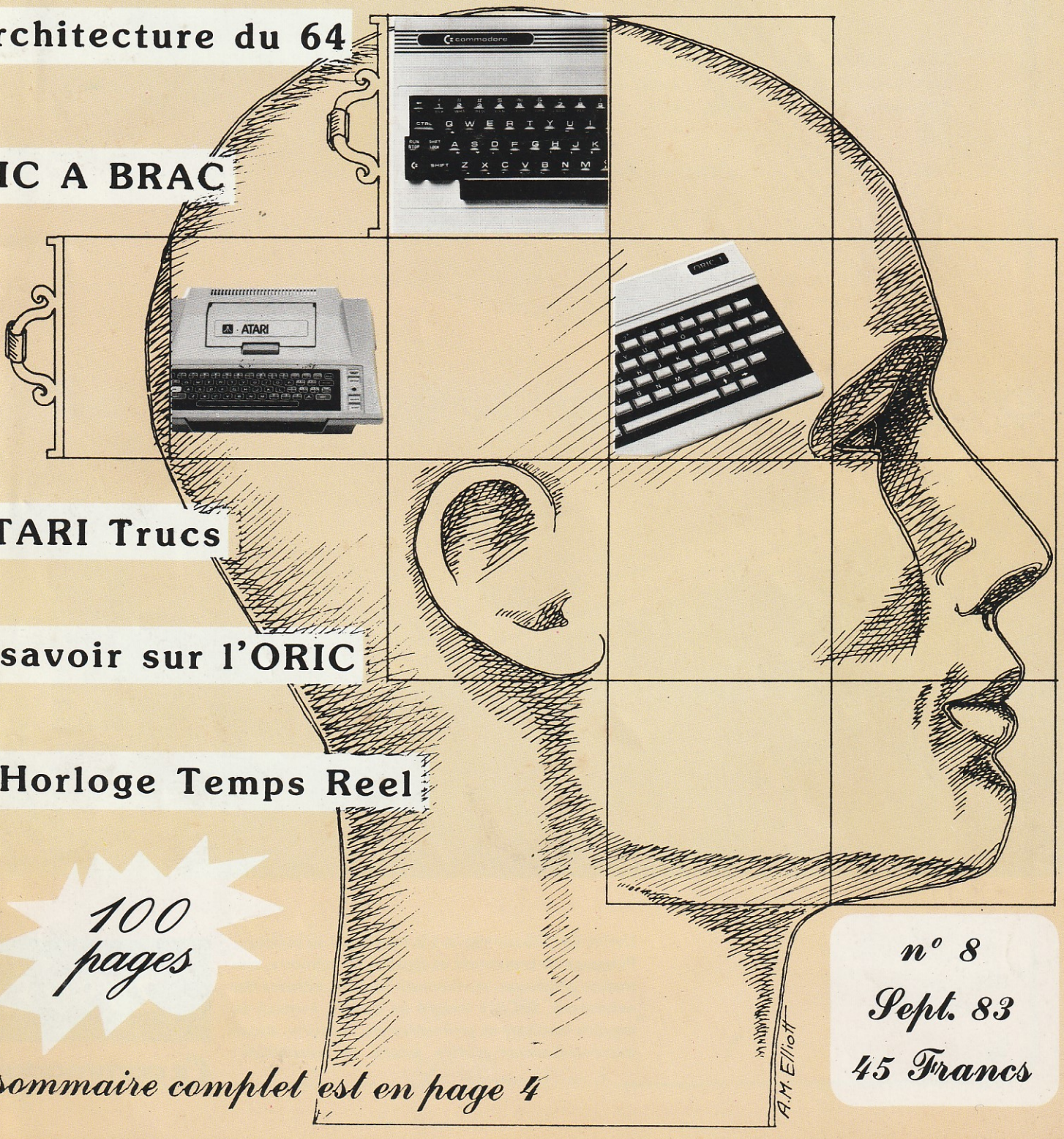
Architecture du 64

VIC A BRAC

ATARI Trucs

A savoir sur l'ORIC

L'Horloge Temps Reel



100 pages

n° 8
Sept. 83
45 Francs

Le sommaire complet est en page 4

COMMODORE 64. L'EXTRAORDINATEUR.

CONCURRENCE MODERNE

L'effet 64 ! Découvrez-le à la maison ou au bureau... Puissance de traitement et capacité de mémoire, animation graphique, synthétiseur musical, richesse des extensions, SECAM intégré en option, éventail de logiciels familiaux et professionnels. A ce prix, aucun micro-ordinateur n'offre autant de possibilités !

Extraordinaire :
2 990 F.
Prix public conseillé.

commodore

EDITORIAL

De retard en retard, et avec la fermeture des imprimeries, c'est en définitive à la date du n° 9 que paraît ce n° 8. Nous espérons maintenant paraître à la date.

Pour compenser, et comme l'annonçait la page du 1er avril (ce n'en était donc pas un), vous avez un numéro plus gros et qui accueille de nouvelles machines.

Bienvenue aux Ataristes et Oriciens !

La Commode se décloisonne. Elle reste spécialisée mais n'est plus exclusive. Quelques inconditionnels de Commodore seront un peu déçus, mais ils verront vite que tous ont à y gagner. D'abord plus de lecteurs, donc plus d'idées, et toute idée développée sur une machine est ADAPTABLE à la vôtre.

Bien sûr, ce n'est pas toujours tout cuit, mais rien d'efficace ne l'est. La Commode veut vous aider dans vos efforts, mais elle ne peut supprimer la nécessité des efforts. Nous essaierons, à chaque article, de montrer les points où des besoins d'adaptations se font sentir et nous en donnerons les lignes directrices. A ce propos, tout lecteur qui aura adapté un programme d'une configuration à une autre ou d'une marque à une autre, est prié de nous envoyer son adaptation pour le plus grand profit des autres lecteurs.

Maintenant, y a-t'il tant d'adaptations que cela à faire ? Les Commodore, Atari et Oric ont tous à peu près la même architecture (ils ont copié, quoi, y a pas de doute!) et ils utilisent tous le 6502 et beaucoup de concepts de leur système et de leur BASIC sont communs. Dans notre numéro 9, commencera notre cours d'Assembleur, commun à tous les modèles. Il utilisera une nouvelle cassette La Commode, l'Assembleur Universel.

Pour que nos abonnés fidèles n'aient pas moins de pages consacrées aux Commodore, nous sommes passés à 96 pages, pour une augmentation de prix bien moindre que les 50% de supplément d'informations. De plus, nous n'avions pas augmenté depuis 2 ans, ce n'était plus tenable !

La Commode vous promet d'être toujours plus riche et toujours plus utile. Le décloisonnement et l'augmentation du nombre de pages sont une étape dans cette voie.

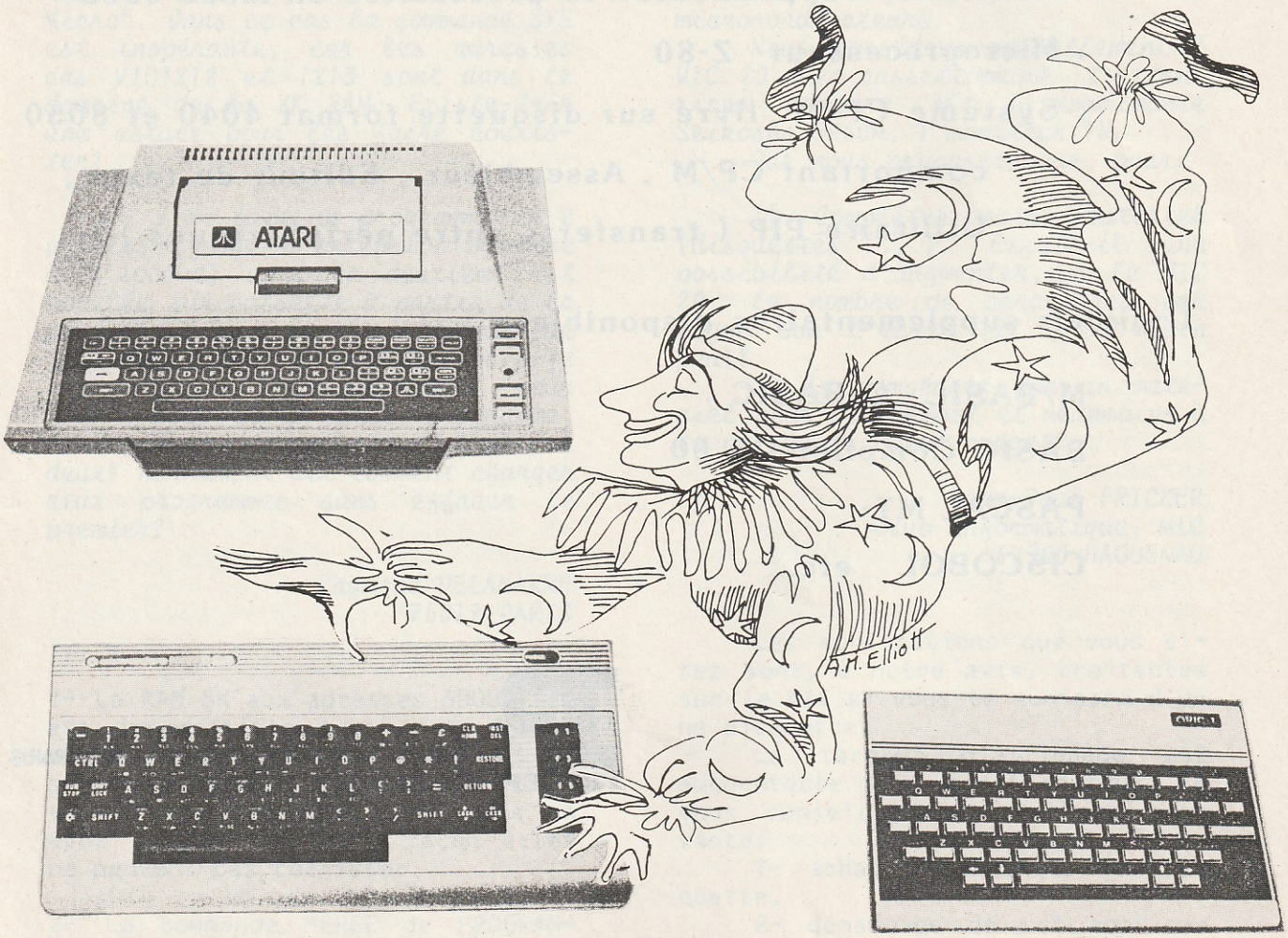
Daniel-Jean DAVID.

Sommaire n° 8

		CBM	VIC /64	ATARI	ORIC
Editorial	3	*	*	*	*
Sommaire	4	*	*	*	*
Courrier des Lecteurs	7	*	*	*	*
Magazine	12	*	*		
Oric à Brac	16				*
Premier contact avec l'ATARI 1200 XL	18			*	
L'horloge Temps Réel sur Commodore, Atari, Oric	20	*	*	*	*
VIC à Brac	23		*		
Routine GET	25	*	*		
Les Ficelles	27	*	*		
Définition d'O.G.D. multicouleurs	29		*		
Atari Trucs	33			*	
64 à Brac	34		*		
Premières notions sur l'architecture du 64	35		*		
Chaînage de 2 programmes	38	*			
Inflation	39	*	*	*	*
Calcul du déterminant d'une matrice carrée	47	*	*	*	*
Routines-Système du 64	49		*		

		CBM	VIC /64	ATARI	ORIC
A la Recherche des Routines Perdues	53	*	*		
BAM BAM BAM	57	*			
Un CBM secrétaire de Laboratoire	64	*			
Flexion 2.5	80	*			
Dessin Animé sur CBM 3032-Edex	86	*			
CLUBS	93	*	*	*	*
Bibliographie	94	*	*	*	*
Réservé aux Abonnés	96	*	*	*	*

La loi du 11 mars 1957 n'autorisant, aux termes des alinéas 2 et 3 de l'article 41, d'une part, que « les copies ou reproductions strictement réservées à l'usage privé du copiste et non destinées à une utilisation collective », et d'autre part, que les analyses et les courtes citations dans un but d'exemples et d'illustrations, « toute représentation ou reproduction intégrale, ou partielle, faite sans le consentement de l'auteur ou de ses ayants-droit ou ayants-cause est illicite » (alinéa 1er de l'Art. 40). Cette représentation ou reproduction, par quelque procédé que ce soit, constituerait donc une contre-façon sanctionnée par les Art. 425 et suivants du Code Pénal.



ELARGISSEZ

L'HORIZON DE VOTRE CBM 4000#

AVEC CP/MAKER

CP/Maker se présente sous forme d'une carte à installer à l'intérieur du CBM offrant les possibilités suivantes

. 64 K RAM supplémentaires accessibles en mode 6502

. Microprocesseur Z-80

. Système CP/M livré sur disquette format 4040 et 8050

- comportant CP/M , Assembleur , Editeur de texte , Utilitaire PIP (transferts entre périphériques)

Logiciels supplémentaires disponibles:

M/BASIC C/BASIC

BASIC Compiler CB 80

PASCAL MT-

CISCOBOL etc...

19-21, RUE MATHURIN-RÉGNIER - 75015 PARIS
TÉL. : 306.82.02 - TÉLEX : 204 875 F.

Valable également pour CBM 8032

Courrier des Lecteurs

De la RAM sur les ROM

J'ai déjà les 6 premiers numéros de La Commode qui ont débroussaillé quelques questions concernant l'utilisation de mon VIC 20. Hélas pas toutes, car il me reste les problèmes suivants:

- Comment utiliser les cartouches Programmer's Aid VIC1212 et Moniteur Langage Machine-VIC1213 avec les extensions 8K RAM-VIC1110, le commutateur de cette dernière étant dans la position (compatible avec la 16K RAM) où ses adresses mémoire vont de $6000 à $7FFF, le tout branché sur l'extension "Arfon Micro". Dans ce cas la commande SYS est inopérante, car les adresses des VIC1212 et 1213 sont dans le domaine de la 8K RAM. Existe-t-il une astuce pour les faire coexister?

- J'ai fait un programme en 2 parties (faute d'extension mémoire à l'époque) dont la deuxième est appelée sur cassette à partir de la première. Comment, maintenant que tout peut tenir en mémoire, puis-je les charger successivement pour n'en faire qu'un seul programme, sans réécrire complètement l'un des deux? Autrement dit comment charger deux programmes sans effacer le premier?

Raymond DELABARRE
75017 PARIS

1- La RAM 8K aux adresses 6000-7FFF est incompatible avec toute ROM à ces adresses. Une solution est de stocker les ROM sur disque et de les charger en RAM quand vous en avez besoin. De toute façon elles ne peuvent pas coexister.

2- La commande MERGE de PROGRAMMER'S AID résout votre problème. Le second programme doit éventuellement avoir été renuméroté.

Gestion d'un club

Notre club informatique est essentiellement tourné vers l'initiation. D'une part avec des rencontres hebdomadaires d'apprentissage avec moniteur, d'autre part avec des manipulations individuelles durant toute la semaine. Par ailleurs le club désire apporter une participation plus active à la vie de notre association, la MLC-centre socio-culturel: gestion du fichier des membres, comptabilité, service de documentation sur les micro-ordinateurs.

Nous possédons actuellement 2 VIC 20 avec cassettophone, 2 extensions mémoire 3K, 1 imprimante Seikosha GP80M, 1 moniteur TV.

Que nous proposez-vous, pour:

1) Compléter notre matériel (disquettes ...). Existe-il une possibilité d'augmenter, sur le VIC 20, le nombre de caractères par ligne sur l'écran (22 c'est bien peu)?

2) Le remplacer par un matériel plus performant et néanmoins à la portée de notre budget?

Jean-Paul FRICKER
Club informatique, MLC
67500-HAGUENAU

Les applications que vous citez sont, à notre avis, traitables sur le VIC si vous le munissez d'une disquette.

La largeur d'affichage est augmentable mais c'est coûteux. Je vous conseillerais la chose suivante:

1- achat d'une unité de disquette,

2- démarrage de vos applications sur VIC.

3- achat d'un Commodore 64 (qui accepte les mêmes disquettes). Transferts de vos applications sur le 64 (ce qui devrait être immédiat). Utilisation du VIC pour les initiations.

Un truc sur INPUT

Dans le courrier des lecteurs du numéro 4, vous avez répondu à ma question sur le brochage d'un 3016 pour extension mémoire d'un 3008.

Je tiens à compléter votre réponse.

Il faut non seulement enlever E et mettre F, mais:

enlever A, E, L, M,
ajouter F, M, I, C.

En mettant des 4116 à la place des 4108 on obtient 16K RAM.

Ce n'est pas moi qui ait inventé ce brochage, mais le magasin "Informatique Center" Lille à qui je dois cela et que je tiens à remercier ici. "Ils" ont été un des rares à répondre à mes nombreuses lettres envoyées pour des renseignements sur l'extension mémoire.

Je tiens à dire que 3 lettres à Procep sont toujours sans réponse. Merci l'importateur!

Ceci dit, voici un petit truc sur INPUT: si on fait INPUT A,B,C le CBM répond "?" et attend 3 réponses. Si on fait RETURN, on sort du programme (ça c'est bien connu), mais si on ne donne qu'une seule réponse, le CBM marque ?? et attend les autres.

Ici, si on répond RETURN tout court, on ne sort pas du programme (0 par défaut), exactement comme le VIC 20. Il reste à découvrir comment passer à la deuxième étape (?) sans passer par la première. Les INPUT seront donc protégés efficacement (un SYS ... sans doute).

Ceci dit, pensez au 64, le bien mystérieux engin. Le manque d'information est étrange...

E. LEGENDRE
21121 DAIX

Ce numéro devrait étancher votre soif d'information sur le 64.

*
* *

80 colonnes sur 4000

Parmi vos projets ne pourriez vous inclure le passage en 80 colonnes des 4000-G ? C'est possible, j'y suis passé par mégarde quand j'ai fait l'extension 32 K. Seulement je n'avais plus, évidemment, que la moitié de l'écran en hauteur ! Naturellement, vu que le BASIC n'est pas prévu pour, il faudrait un inverseur (manuel ou à relais sur le VIA) pour ne l'utiliser que pour l'affichage pour les grands tableaux par exemple.

Avez-vous aussi songé à nous décrire un stylo lumineux ? Ce petit objet rendrait bien des services.

Jean-Paul LAURENT
13010 MARSEILLE

Le stylo lumineux a été traité dans le numéro 7.

La question des 80 colonnes sera traitée dans un prochain n°.

Simple Software

La naissance et le développement de votre revue viennent enfin combler un grand vide dans la presse spécialisée française. Les rubriques régulières ou les articles consacrés à la gamme Commodore des magazines l'Ordinateur Individuel ou Micro-Systèmes sont, malgré leur intérêt, insuffisants pour un possesseur de matériel Commodore. Le bulletin édité par Procep avait le défaut d'émaner de l'importateur et donc de ne pas être suffisamment critique à l'égard du matériel et du logiciel.

Permettez-moi avant de conclure de recommander aux lecteurs de La Commode deux programmes d'origine américaine que j'utilise régulièrement et dont le rapport qualité/prix est excellent. Le premier est un programme de gestion de fichier "Simply File" (prix: 65 livres anglaises soit environ 750 FF), le second est un programme de traitement de texte qui a l'avantage de pouvoir fonctionner sur cassette et/ou sur disquette "Simply Write" (prix 40 livres sur cassette et 45 sur disquette, soit respectivement environ 460 FF et 520 FF). Ces deux logiciels ne sont pas, à

ma connaissance, distribués en France. On peut les commander auprès de la maison :

SIMPLE SOFTWARE LTD
15 Havelock Road
BRIGHTON SUSSEX BN1 6GL
GRANDE BRETAGNE

Le directeur de cette S.C.I., M. Brian Tregar, applique des règles de commercialisation qui mériteraient les félicitations de la plus sévère des organisations de consommateurs.

Musique

Informaticien de loisir, novice, seul à la campagne, possesseur récent d'un CBM 4016 G.Ecran, je suis bien sûr confronté à de nombreux problèmes. Je vous en soumets un :

Voulant apprendre la musique à cet outil, j'ai décidé que la rangée des touches inférieures serait les notes (do,ré,mi,etc,). Celle immédiatement supérieure les dièzes. Connaissant la valeur du PEEK, des touches pressées (en faisant :

```
10 P = PEEK (151)
20 GET A$
30 IF A$ = " " THEN 10
40 PRINT P : GOTO 10
```

Il suffit alors de rentrer le programme suivant:

```
10 P = PEEK (151)
20 J = 59464
30 IF P = 90 THEN  I = 242 (do) :
GOSUB 1000
40 IF P = 88 THEN I = 215 (ré) :
GOSUB 1000
etc pour toutes les notes.

1000 : POKE J + 3,16 : POKE J +
2,15 : POKE J, I : RETURN.
```

Hélas, (trois fois) si l'ordinateur accepte bien de jouer toutes les notes programmées avec son petit haut parleur interne et change bien de note en fonction des touches programmées, il refuse de reconnaître que la touche n'est plus pressée et la dernière note continue indéfiniment..., J'ai dû ajouter en 999 : IF P = 32 THEN POKE 59467, 0 : GOTO 10 pour que la touche "espace" arrête le son.

Il doit bien y avoir un moyen pour que le son programmé commence quand on appuie sur une touche et s'arrête quand on la lâche. Oui, mais lequel ?

Paul BOULAY
32130 SAMATAN

La solution de votre problème est très simple.

Trouvez la valeur X du PEEK (151) lorsqu'il n'y a pas de touche enfoncée à l'aide du programme :

```
10 ?PEEK(151):GOTO10
```

Normalement, c'est X = 255
Ajoutez alors à votre programme :

```
25 IF P=X THEN POKE 59467,0:GOTO 10
```

(Remarque : si à chaque ligne de votre programme vous ajoutez un : GOTO 10 derrière le GOSUB 1000, vous améliorerez le temps de réponse)

Statistiques

Je suis débutant, j'ai acheté un VIC 20 et j'aimerais faire un petit programme basé sur les nombres aléatoires; j'ai 20 étalons et je voudrais en choisir trois, je ne sais comment élaborer le programme.

Victor TOURRE
83100 TOULON

Autant que nous comprenions votre problème, il s'agit de tirer au sort, parmi 20, les trois numéros N1; N2 et N3 des étalons que vous voulez choisir. La solution est alors très simple :

```
N1 = 1 + INT (20 * RND (1) )
N2 = 1 + INT (20 * RND (1) )
N3 = 1 + INT (20 * RND (1) )
```

Vous pouvez vérifier les numéros obtenus par ? N1, N2, N3.
Si de plus N1, N2 et N3 doivent être différents, vous devez reprendre le calcul de N2 tant que N2=N1, et vous devez reprendre celui de N3 tant que N3=N1 OU N3=N2.

*
* *

Un manuel 8050 ?

Je vous serais très reconnaissant de bien vouloir m'indiquer s'il existe un manuel concernant l'unité de double disquette modèle 8050 Commodore.

Philippe BOUCHET
86180 BUXEROLLES

A part quelques renseignements qu'on peut trouver dans La Commode, l'unité CBM 8050 est traitée dans le livre de PROCEP Manuel de l'utilisateur du CBM 8001 (éditions Radio) et dans D.J. DAVID Pratique du PET/CBM vol. 1 : périphériques et gestion de fichiers. (Ed. du P.S.I.)

Plus de 255 caractères programmables

Je vous écris "en urgence", car j'ai un problème que je n'arrive pas à résoudre. Possesseur d'un VIC 20 avec extension 16 K, je réalise une étude sur le Languedoc-Roussillon puis le Gard, en utilisant abondamment le graphisme haute résolution pour le tracé des cartes, diagrammes, etc., Je n'ai pas eu de problèmes jusqu'au jour où j'ai dépassé 255 caractères . Je procède comme l'indique PROCEP en réorganisant la partition mémoire en ramenant la RAM écran en 7680 et la RAM couleur en 38400. Ceci me laisse donc pour la haute résolution la possibilité de définir 448 caractères (de 4096 à 7680).

Bien entendu, comme les cartes visualisées sont accompagnées de caractères alpha-numériques, je conserve systématiquement les 128 premiers caractères du générateur (de 4096 à 5119).

Question : Comment faire pour appeler les caractères, par moi définis, dont le numéro est au-delà de 255...jusqu'à 448 (POKE 7800, x avec x 255 !!)

Roger BOUISSET
30000 NIMES

Des éléments de réponse à votre question ont été donnés dans La Commode n° 5 p.35.

On ne peut pas définir plus de 256 caractères programmables, puisque le code doit tenir en un octet. Mais vous pouvez couvrir tout l'écran en appliquant la technique du doublement des mailles esquissée dans La Commode n°5, p. 36.

17 questions

Je possède un VIC 20 version de base + magnéto depuis presqu'un an et demi et me trouve un peu bloqué. Ne parlons pas du manuel d'utilisation (en anglais au moment de l'achat) qui est surtout un B.A.BA. Heureusement, il existe : La découverte du VIC de Daniel-Jean DAVID (que je salue au passage), livre extrêmement bien fait, qui remplace avantageusement le manuel d'auto formation basic (75 FF au lieu de 250 FF environ) mais qui laisse sur sa faim vers la fin. A ce sujet, il y aura de quoi dévorer dans le courant du 1er trimestre 1983. La suite est à paraître. Du même auteur, même collection (P.S.I). En attendant, je souhaiterais des renseignements sur certains points :

1 - Pourriez-vous publier un schéma de principe imprimé et un schéma d'implantation du VIC 20.

2 - Pourriez-vous publier une liste des adresses et de leurs affectations complète et dans l'ordre (voir n°3 et 4 de La Commode).

3 - Pourriez-vous aussi publier une liste des commandes Basic et langage machine.

4 - Comment accéder au langage machine?

5 - Quelles sont les possibilités du VIC en traitement de texte et éditeur ?

De plus, certains points restent obscurs :

6 - Est-il possible d'effacer un programme sans effacer les variables précédemment incrémentées par ce programme?

7 - Comment effacer une longue partie d'un programme sans

taper les n^os de lignes un par un suivi de RETURN ?

8 - Est-il possible de charger un programme à la suite d'un autre sans effacer le premier. Si oui, comment ?

9 - Est-ce qu'un programme peut travailler sur un autre programme (renumérotation, par exemple) ?

10 - Peut-on travailler sur les messages d'erreurs ? Quels sont leurs codes particuliers?

11 - Est-il possible d'indicer une variable alphanumérique dans un ordre INPUT exemple (INPUT A$ (I)) et dans l'affirmative est-il correct de faire 100 A$ (I + N) = B$?

12 - Est-il possible de générer des caractères ? Si oui, comment ?

13 - Et, enfin, est-il possible de faire défiler l'écran vers la gauche, la droite, le haut, le bas, à volonté ?

14 - Pourriez-vous donner une liste des logiciels disque,cassette et cartouche existants ou à venir ?

15 - A quoi correspondent les instructions :
10 A = B AND 7
10 A = B:A = 7 ?

16 - D'ordre plus général, quelles sont les différences entre le VIC 20 et VIC 40 ?

17 - De plus, j'ai lu dans la revue ECHOS SINCLAIR, page 22, que SINCLAIR va commercialiser un lecteur de mini-disquettes d'une capacité de 100 K octets pour environ 600 FF. Etes-vous au fait d'un tel système ? Sera-t-il compatible avec le VIC ou d'autres matériels COMMODORE ? Quelle est la politique de COMMODORE au sujet du développement de petits systèmes personnels ?

Pierre FREMION
75013 PARIS

1 - Seul COMMODORE est en droit de publier un tel schéma.

2 - Nous n'avons pas mieux que les listes publiées dans les Nos 3 et 4. Il y aura de plus en plus d'exemples d'utilisation dans la Commode.

3/4 - cf. La Découverte de VIC et La Pratique de VIC vol.II qui va bientôt paraître.

5 - Il existe plusieurs systèmes de traitement de texte sur VIC. PROCEP en importe un, voyez ses distributeurs.

6 - Impossible.

7/8/9 - Résolu par Programmers's Aid.

10 - Un message d'erreur est imprimé par chargement dans X du numéro de l'erreur et appel d'un sous-programme général d'impression du message.

11 - Oui.

12 - Oui. Voyez, entre autres, Vic à Brac du n°7.

13 - Haut / bas, facile
Droite/gauche, très difficile (beaucoup plus facile sur 64)

14 - Voir catalogue PROCEP pour ceux qui existent. Pour les autres, demandez au Professeur PHAK-KHIR.

15 - La 1ère isole les bits 0 à 3 de B (reste de la division de B par 8).
La 2ème ligne transfère B dans puis donne à A la valeur 7.

16 - Le VIC 20 existe; pas les autres.

17 - Il existe une interface VIC imprimante SINCLAIR dont nous parlerons bientôt plus longuement. La même chose peut se faire sur les disques.

*
* *

MAGAZINE

Visite au PET SHOW

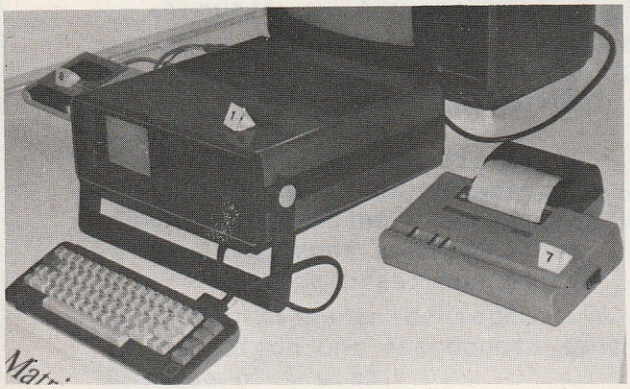
Le 4ème PET SHOW s'est déroulé à Londres du 9 au 11 juin.

La foule se pressait...

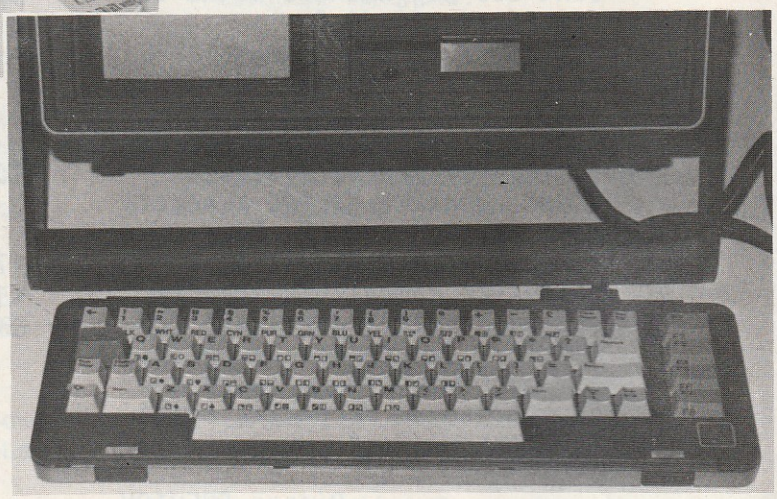
pour voir les logiciels...

... et les nouveaux matériels Commodore.

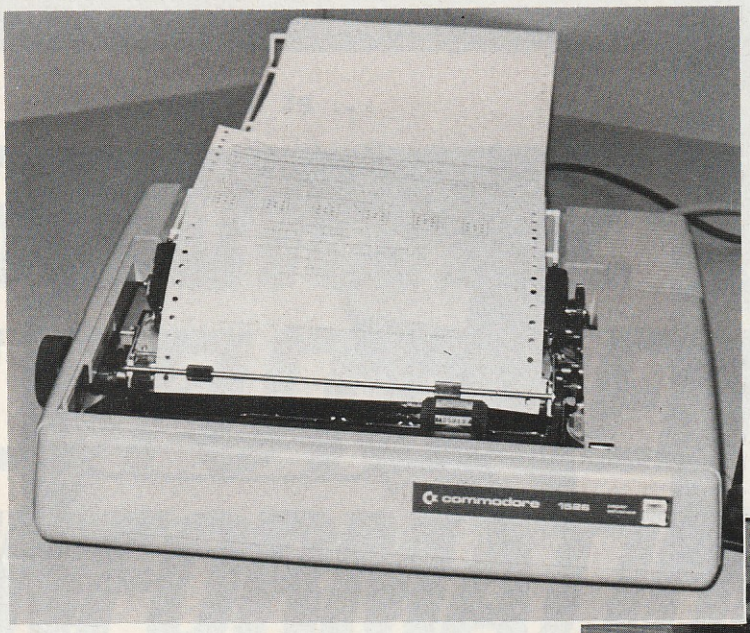
le 64 portable
le moniteur couleur
le 8000 SK
le 700
l'imprimante 6400
la 8250 bas profil



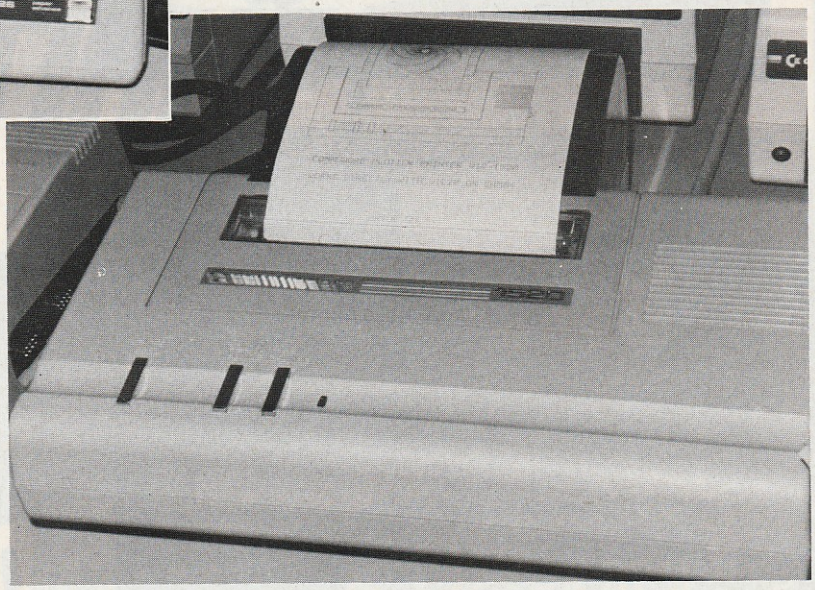
le 64 portable



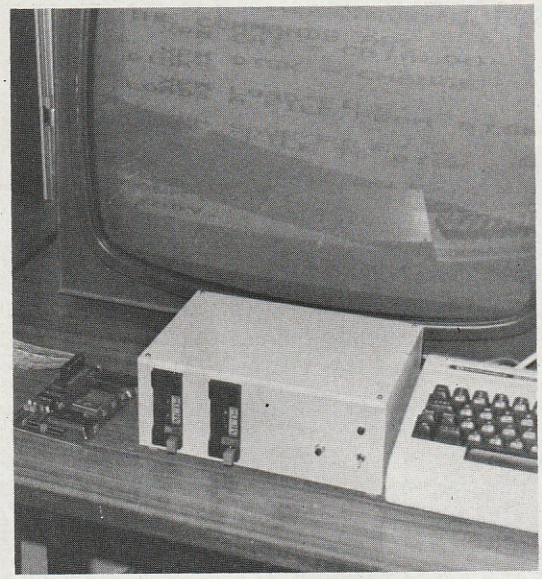
gros plan sur le clavier, la poignée, l'unité de disques.



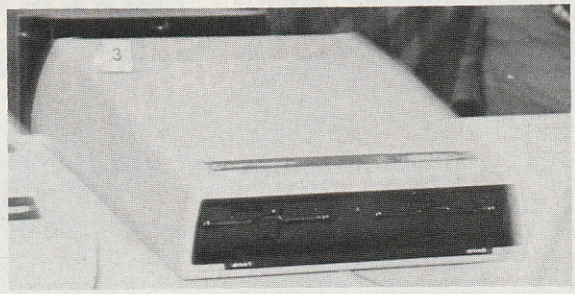
Une imprimante plus performante pour le 64 : la 1526



L'imprimante traceur 1520 pour VIC et 64



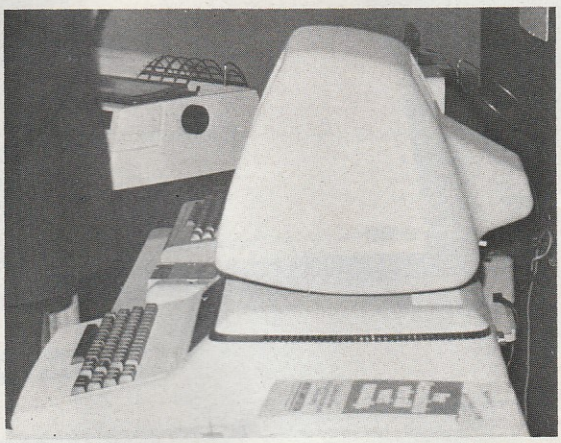
Des disques 3 pouces pour VIC et 64

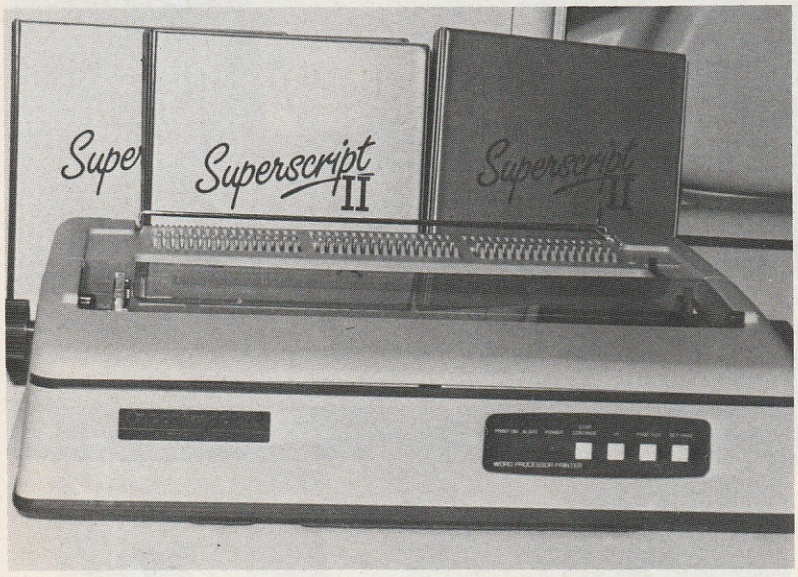


L'unité 8250 bas profil :
une forte capacité pour un faible encombrement.

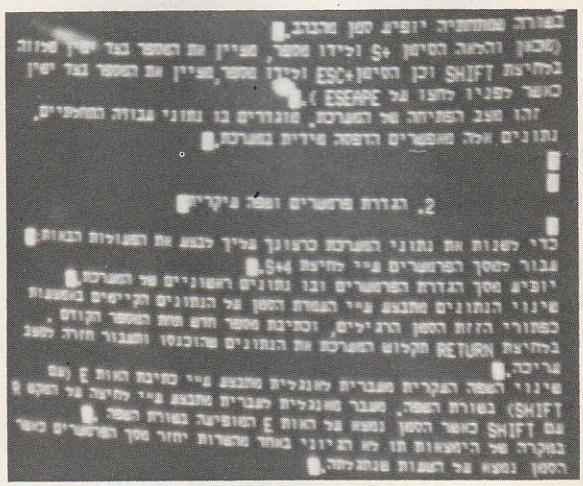
Le 700 : de face ...

... et de profil





Des systèmes de traitement de textes : SUPERSCRIPT avec la nouvelle imprimante à marguerite CBM 6400...



... et ED, qui aurait pu nous servir pour l'éditorial du n° 7.

La plus grande leçon de ce PET show est l'incursion que fait le 64 dans le domaine des applications professionnelles. Il semble d'ailleurs que les développeurs de logiciels écrivent simultanément leurs programmes pour 700, 8000 et 64. C'est le cas de SUPER BASE de SUPER SCRIPT et de CALC RESULT.

Un autre élément est l'arrivée de nouveaux logiciels de traitement de textes très intéressants, notamment ED et SUPER SCRIPT. A quand leur francisation ? SUPER SCRIPT se couple à SUPER BASE pour former SUPER OFFICE, non concurrent dangereux de SILICON OFFICE.

Au point de vue hardware, gros intérêt des visiteurs pour le 64 portable, que nous annoncions déjà dans notre numéro précédent. Sinon, plus un mot du MAX, la machine qui devait compléter la gamme vers le bas. Le VIC en est tout ragaillardi et, contrairement à ce que disent certains de nos confrères, nous ne pensons pas qu'il soit arrêté de sitôt, ce qu'il risquait beaucoup plus s'il avait dû être en sandwich entre le MAX et le 64.

Daniel-Jean DAVID.

BRANCHEZ-VOUS SUR LE FUTUR EN MICRO INFORMATIQUE !

SICOB BOUTIQUE

CNIT-PARIS LA DEFENSE
DU 21 AU 30 SEPTEMBRE
DE 9 H 30 A 18 H. FERME LE DIMANCHE 25
ENTRÉE LIBRE

X429

PUBLICIS

ORIC à BRAC

A Savoir sur l'ORIC

Quelques curiosités de l'ORIC

. 10 GET R$: PRINT R$,ASC (R$) donne pour l'apostrophe une ?ILLEGAL QUANTITY ERROR IN 10 alors que PRINT ASC (" ' ") donne bien 39.

. TAB (x)
- jusqu'à x = 13 début impression sur 1ère colonne du texte ensuite, début à x-13,
- tabulations successives fantaisistes, ex :
  PRINT TAB(15)"1"TAB(20)"2" TAB(25)"3" TAM(30)"4" donne 1 en col. 2, 2 en col. 10, 3 en col.23 et 4 en col. 3 de la ligne suivante.

. Utilisez plutôt SPC qui, elle, marche bien.

Il existe 2 caractères ASCII non mentionnés :

126 = (par POKE ou CHR$)

127 = ■ (par POKE seulement ou par CHR$(255)
(également code ASCII de DEL)

- à noter aussi c, code ASCII 96 qui ne peut être obtenu au clavier, mais par POKE ou PRINT CHR$ (96).

En cas de répétition d'un caractère obtenu avec SHIFT on peut relâcher cette touche après la première impression ; on reste en mode shifté.

. Un conseil : pour toutes les instructions ayant une action sur la mémoire (POKE, PEEK, PLOT, etc...) donner des valeurs en hexa et non en décimal, temps d'exécution dans le rapport 1 à 1,7.

. Le dernier argument des fonctions graphiques, appelé FD (fond-devant) peut prendre les valeurs 0-1-2-3 :
0 = couleur du fond (papier)
1 = couleur du devant (encre)
2 = couleur inversée
3 = pas de couleur

En fait, 2 produit la même action que 1.

. Il y a dans la table des mots clés 2 instructions :
- INVERSE,
- NORMAL,

qui à l'exécution donnent :
SYNTAX ERROR ????

Jean RENAUD

Chargement et sauvegarde sur Cassette

L'ORIC a deux vitesses de gestion de la cassette : 2400 bd et 300 bd pour une fiabilité accrue.

Si un chargement n'arrive pas à se faire, vérifiez :

a) la commande utilisée :
CLOAD"" pour 2400 bd,
CLOAD"",S pour 300 bd.

On ne peut lire à une vitesse un programme écrit à l'autre vitesse.

b) le branchement utilisé, revérifiez avec le manuel (p 7).

c) le règlage du volume. En principe, l'optimum est à 3/4 du maximum, mais il faut faire des essais (c'est l'ennui de l'utilisation des magnétos ordinaires). Si le niveau est trop bas vous pouvez avoir le diagnostic LOAD ABORTED. Si le niveau est trop haut, vous pouvez avoir des lignes de U au lieu du programme.

Rappelons que le magnéto utilisé n'a pas besoin d'être HIFI.
Pour le nom, utilisez des " et non des '.

Fonction VAL

Si VAL renvoie 0, c'est qu'il y a un caractère de contrôle illégal dans la chaîne (codes de 0 à 31).

Commande FILL

Elle ne met pas automatiquement à jour la position du curseur. En cas de besoin, utiliser CURSET après le FILL.

POKE HEXA

La donnée à écrire par POKE ne peut être en hexadécimal (#...). C'est un manque de cohérence car l'adresse peut l'être.

Chargement et code machine

On ne peut pas charger un programme machine à partir d'un programme BASIC.

Il faut d'abord protéger la zone mémoire de BASIC par la commande HIMEM.

De même, si vous utilisez le jeu de caractères auxiliaire, protégez-le de BASIC par HIMEM #9FFF au début de votre programme.

GET

La fonction GET doit être employée uniquement avec une variable chaîne de caractères : elle prendra un caractère. Mais ce n'est pas une saisie au vol : GET attendra la venue du caractère.

Pour la saisie au vol, utiliser la fonction KEY$.

Port CENTRONICS

L'interface Centronics envoie la séquence retour-chariot + alimentation-ligne. Si l'alimentation ligne est de trop, elle doit pouvoir être inhibée sur l'imprimante.

Fonction KEY$

Elle permet la saisie au vol de caractères. Si vous maintenez l'appui sur une touche, la fonction ne décèlera qu'un appui : elle décèlera un second appui au bout du délai de répétion de caractères.

Donc, si vous voulez faire une cascade de tests, il faut faire :

```
10 A$ = KEY$          20 IF KEY$ = ..
20 IF A$ = ..et non   30 IF KEY$ = ..
30 IF A$ = ..                 "
```

Honoric de BALSA
(pseudonyme un peu ... léger)
(c'est promis, on n'en fera plus!)

Premier Contact avec l'ATARI 1200 XL

La nouvelle ligne de produits ATARI s'appelle XL. Elle sera formée au départ du 1200 XL et du 600 XL. Le 1200 XL est déja apparu aux Etats Unis. Il y aura ensuite les 800 XL, 1400 XL et 1450 XL.

La nouvelle série de micro-ordinateurs ATARI s'appelle XL. On en connaît 2 membres, le 600 et -seul sorti actuellement aux U.S.A.- le 1200. La nouvelle série se caractérise par une compatibilité logicielle complète avec les 400/800.

Le 1200 XL est un développement du 800. Il apparaît comme un concurrent du 64. Nous ne connaissons pas son futur prix en France. En tous cas, c'est, lui aussi, une machine de 64 K de RAM, toujours à 6502, donc son architecture ne peut qu'être semblable à celle du 64 et de l'ORIC (cf. dans ce numéro l'article sur l'architecture du 64) : une partie des 64 K de RAM est occultée pour laisser place à la ROM et aux E/S. Les différences se font au niveau des détails sur la façon d'effectuer cette occultation, pas au niveau des principes.

Le clavier

L'aspect général de l'ATARI 1200 XL est, à notre avis, très amélioré par rapport au 800 qui fait un peu massif mais est très robuste. Le 1200 est très fin, très plat. Les fentes d'aération sur les côtés sont fictives : elles donnent un aspect APPLE dont nous ne comprenons pas la raison.

Le 1200 XL présente un clavier équivalent dans sa fonctionnalité à celui du 800, mais des améliorations ont été apportées sur l'emplacement de certaines fonctions et certaines touches ont été ajoutées : il y a maintenant 56 touches classiques plus 11 touches spéciales au-dessus du clavier.

Les 11 touches spéciales comprennent :

- les 4 touches RESET, START, SELECT et OPTION qui étaient à part, à droite du clavier du 800,

- 4 touches de fonction F1 à F4, les touches "vidéo inverse" (ATARI) et BREAK qui étaient incorporées au clavier du 800. Nous approuvons le fait que ces touches -dont la fonction est très spéciale- aient été mises à part.

- la touche HELP, qui est nouvelle. Elle a deux fonctions :

  . elle peut servir à afficher des instructions dans un programme, sur demande d'un utilisateur en détresse

  . elle sert de touche de diagnostic, lançant un programme qui vérifie les connexions et le fonctionnement des principaux sous-ensembles.

Les touches de fonction ont des rôles très intéressants :

. l'une d'elles met le clavier en "mode européen", ce qui permettrait facilement d'avoir un clavier AZERTY. Avis aux auteurs de systèmes

de traitement de textes.

. une autre désactive le clavier ce qui évite qu'un programme ne soit troublé par une frappe intempestive,

. une autre désactive l'affichage ce qui évite l'usure du tube si la machine reste longtemps inactive. Le 400/800 obtenait le même résultat en changeant les couleurs lorsqu'on restait longtemps sans taper.

L'organisation du clavier principal a été améliorée. Les deux SHIFT sont larges. Les touches CRTL, TAB, ESC, CAPS/LOWER et DELETE/BACK SPACE ont été élargies. Les touches de mouvement de curseur agissent maintenant sans CTRL (une sérieuse amélioration).

L'appui sur les touches n'est plus accompagné d'un signal sonore, ou plutôt, il est produit par la télé, donc on peut l'atténuer. En plus du voyant "sous tension", il y a deux voyants supplémentaires : "clavier inhibé" et "mode européen".

Autres nouveautés de l'unité centrale

Il n'y a plus qu'un emplacement pour cartouche de ROM (comme le 400, alors que le 800 en a deux) et il est maintenant sur le côté : il est vrai qu'il n'y avait eu qu'une cartouche produite pour aller dans le second emplacement.

De même, le nombre de connecteurs pour poignées de jeu a été réduit de 4 à 2, ce qui est suffisant alors que les 4 étaient un peu superflus encore que certains jeux ATARI les utilisaient. Noter que le 64 est à 2 connecteurs de jeux lui aussi.

Les possibilités de couleurs et de sons des 1200 XL sont identiques à celles du 400/800.

Nouveaux périphériques

Une nouvelle ligne de périphériques est en développement. Ils seront compatibles avec le 400/800 et ils auront une interface 1200 intégrée de telle sorte que le module d'interface 850 qu'il faut avec le 400/800 n'est pas nécessaire.

Parmi ces périphériques, le magnétophone 1010 (le 410 reconditionné), l'imprimante 80 colonnes 1025 et le traceur-imprimante 40 colonnes en couleurs 1020.

Conclusion

L'ATARI 1200 XL semble donc un système séduisant d'autant qu'il sera accompagné de logiciels nouveaux ou anciens réécrits, très adaptés aux applications domestiques: jeux, finances familiales, traitement de textes et apprentissage musical.

Daniel-Jean DAVID

Abonnez-vous à La Commode

L'Horloge Temps Réel sur les Commodore, Atari et Oric

Cet article est le premier d'une série que nous appelons "Trio". Dans chaque article de cette série, on prend un problème général et on compare comment ce problème est traité par les trois marques Commodore, Atari et Oric. Le premier article de la série concerne l'accès à l'horloge temps réel. Les suivants regarderont la saisie de caractères au vol, la modification d'un programme par lui-même, l'accès au Langage-Machine par Basic, le positionnement sur l'écran etc... Ces articles, il va sans dire, seront très précieux pour adapter un programme d'une machine à une autre sans se limiter à une traduction mot à mot.

Fonctionnement de l'horloge temps réel

L'horloge temps réel fonctionne à peu près de la même façon dans les trois marques. Le système est, comme on dit, commandé par interruptions périodiques : tous les soixantièmes de seconde, le processeur est interrompu; il abandonne temporairement sa tâche (qui est en principe de s'occuper de votre programme) pour exécuter des tâches système, notamment examiner le clavier et gérer l'horloge. Gérer l'horloge consiste tout simplement à incrémenter une case mémoire (en fait plusieurs car il faut plus de 8 bits). On voit que toute consultation de ces cases mémoire (par PEEK) donnera le nombre d'interruptions exécutées depuis la mise sous tension ou la dernière remise à 0 des mémoires, donc une valeur proportionnelle au temps écoulé.

Les Commodore et l'Atari utilisent trois cases mémoire, qui sont incrémentées. L'Oric utilise deux cases mémoire et elles sont décrémentées.

Voici les adresses :

	poids faible	poids fort	le plus fort
PET	514/$202	513/$201	512/$200
CBM	143/$8F	142/$8E	141/$8D
VIC/64	162/$A2	161/$A1	160/$A0
ATARI	20/$14	19/$13	18/$12
ORIC	630/$276	631/$277	-

On remarquera que les Commodore et Atari n'obéissent pas ici à la succession habituelle poids faible...poids fort.

Accès sur Commodore

Vous n'avez en fait jamais à accéder aux cases mémoire citées car vous disposez sur Commodore de deux variables réservées TI et TI$. TI est le compte des soixantièmes de seconde : tout se passe comme si TI=PEEK(162)+256*PEEK(161)+4096*PEEK(160)

TI$, elle, donne l'heure sous forme de chaîne de caractères au format HHMMSS.

Initialisation

On initialise un compte de temps à 0 par TI$="000000"

On ne peut pas faire TI=0. En revanche on pourrait faire les trois POKE adresse,0 .

Dans l'utilisation de TI, ne pas oublier qu'on a le temps en 60iè de seconde, donc faire TI/60 pour avoir des secondes et TI/3600 pour avoir des minutes.

Accès sur Atari

La gestion se fait obligatoirement par PEEK aux adresses. On aurait pu penser faire un :

```
DEF FNTI=PEEK(20)+256*PEEK(19)+
4096*PEEK(18)
```

mais DEF FN n'existe pas sur Atari! Il faut donc faire les PEEK.

Les choses peuvent se simplifier en fonction des problèmes à traiter. On peut ne pas avoir des temps trop longs à gérer et ne pas avoir besoin du temps au 60ème de seconde.

Si l'on néglige le PEEK(20), PEEK(19)+256*PEEK(18) donne le temps en groupes de 4 secondes (256 fois un soixantième).

Si l'on néglige le PEEK(18), on ne pourra compter que 256*4 secondes soit un intervalle d'environ 1000s = 20mn. Le temps en minutes sera PEEK(1/15. Le temps en minutes sera PEEK(19)/15. Le temps en secondes sera 4*PEEK(19).

Initialisation

La mise à zéro pour un intervalle se fait par les trois POKE, le seul POKE 19,0 si l'on ne garde que cette case.

Pour mettre à une heure du jour, il faut convertir cette heure en groupes de 4 secondes et en groupes de 1000 secondes et faire les POKE correspondants :

Ex. H heures,M minutes, S secondes
On convertit le tout en secondes par SS=S+60*M+3600*H. Les "quatre secondes " sont Q=SS/4. D'où
T8=INT(Q/256) et T9=Q-256*T8

Enfin POKE 20,0 : POKE 18,T8
:POKE 19,T9 .

Accès sur l'Oric

Comme il y a décomptage, le temps en soixantièmes de seconde est 65535-(PEEK(630)+256*PEEK(631) ou plutôt, puisque l'Oric dispose du "PEEK double" qui s'écrit "DEEK" TI=65535-DEEK(630) .

On peut parfaitement faire :
DEF FNTI(X)=65535-DEEK(630)
Le temps en secondes est :
DEF FNS(X)=FNTI(X)/60
et celui en minutes est :
DEF FNM(X)=FNS(X)/60 .
Rappelons la limitation à des intervalles de 20 mn environ ce qui interdit toute gestion de l'heure dans la journée qui ne soit pas très lourde : votre programme doit déceler les passages à 0 toutes les 20 mn et les compter.

Initialisation

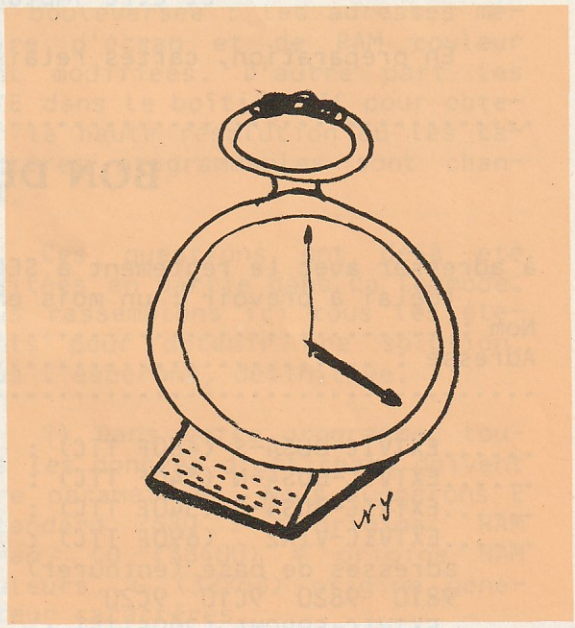
Vous faites deux POKE ...,255 ou plutôt :
DOKE 630,65535
pour initialiser le temps écoulé à 0.

Conclusion

L'horloge temps réel est importante pour beaucoup d'applications : petits intervalles pour les jeux, heure du jour dans certains contrôles de processus. Nous avons vu les techniques à employer dans les trois marques.

Rappelons une limitation : comme la gestion des cassettes détourne les interruptions (et d'autres périphériques peuvent le faire aussi), l'heure devient inexacte après une entrée-sortie cassette (ou autre éventuellement). Il semble que, sur Oric, l'horloge soit aussi perturbée par l'instruction WAIT. Le 64 semble plus à l'abri puisqu'il a un boîtier, le 6526, qui gère l'horloge temps réel.

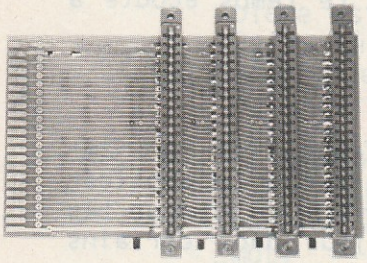
Daniel-Jean DAVID



Pour étendre votre VIC jouez la bonne carte avec SEDERMI

SEDERMI (société éditrice de LA COMMODE) présente un ensemble de cartes d'extensions pour le VIC, caractérisées par un rapport qualité/prix sans concurrence. En préparation, systèmes de cartes pour 64 et ORIC.

Ce sont :

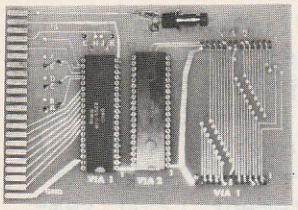


a - Les cartes EXTVIC-BUS 4

qui permettent de connecter simultanément 4 cartouches ou extensions à votre VIC, par exemple VICMON, PROGRAMMER'S AID,SUPER EXPANDER et une extension mémoire, ou toute autre combinaison.

La carte EXTVIC-BUS 4 existe en 3 options :

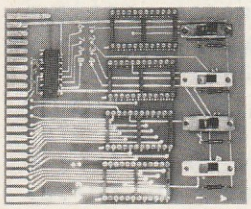
- N La carte montée et testée, avec alimentation par le VIC; c'est l'option normale.
- E Comme N, mais alimentation extérieure (non fournie). Cette option est utile pour certaines extensions d'entrées-sorties et applications particulières.
- S Permettant, grâce à 4 interrupteurs d'activer la cartouche voulue et ainsi d'avoir 4 cartouches de jeu ou Super Expander montées en permanence.



b - Les cartes EXTVIC-VIA 2

qui procurent 40 lignes d'entrées-sorties supplémentaires (idéal pour commander des appareils domestiques, des trains électriques etc.)

Chaque carte contient deux VIA 6522. Il y a 4 adresses de base au choix (à préciser à la commande). On peut connecter simultanément 4 cartes EXTVIC-VIA 2.



c - La carte EXTVIC-EPROM 4

qui contient 4 supports d'EPROM 2532, ce qui vous permet d'avoir vos programmespermanents figés en EPROM et disponibles dès la mise sous tension.
Chaque EPROM peut être activée ou inhibée par interrupteur et être implantée au choix en 4 adresses.

En préparation, cartes relais, programmateur d'EPROM,etc..,

.....

BON DE COMMANDE

à adresser avec le règlement à SEDERMI,28,rue Vicq d'Azir,75010 PARIS
(délai à prévoir : un mois environ)

Nom : .....
Adresse : .....
.....

.....EXTVIC-BUS4-S (480F TTC) : : règlement par
.....EXTVIC-BUS4-N (340F TTC) : : CCP : ()
.....EXTVIC-BUS4-E (340F TTC) : : CB : ()
.....EXTVIC-VIA2 (690F TTC) : : mandat : ()
adresses de base (entourer)
9810 9820 9C10 9C20
.....EXTVIC-EPROM4 (290FTTC) : :
TOTAL ..... F TTC

VIC A BRAC

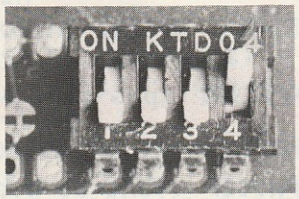
Une mémoire d'éléphant pour votre VIC

De nombreux lecteurs nous ont écrit pour nous manifester leur déception lorsqu'ils connectent à la fois une cartouche 8K (VIC 1110) et une 16 K (VIC 1111) sur un fond de panier (par exemple un EXT VIC BUS 4).

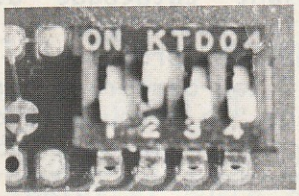
Tout se passe comme s'il n'y avait que la 16 K : on obtient 19000 octets libres. Le fond de panier n'est aucunement responsable. C'est que la 16 K est aux adresses $2000 à $4FFF alors que la 8 K est de $2000 à $3FFF : elle ne rajoute rien.

Il se trouve que la 8 K possède un jeu d'interrupteurs permettant de changer son adresse d'implantation. Il faut amener celle-ci à $6000-$7FFF. Pour cela, ouvrir la cartouche (dévisser la vis puis manoeuvrer d'abord les ergots des fentes arrière avec un tournevis ; enfin louvoyer latéralement jusqu'à ouverture) puis faire passer l'état des interrupteurs :

de



à



Si vous obtenez que votre distributeur le fasse pour vous, ce n'en est que mieux. Le scandale est que beaucoup de distributeurs n'étaient même pas au courant de l'opération (des lecteurs nous l'ont écrit).

Cela vous amène à 28000 octets libres et vous n'aurez jamais plus. Lorsque ces extensions sont présentes, BASIC ne "voit" plus les 3 K. Il faut noter que lorsque la 8 K a été mise en $6000-$7FFF, VICMON et PROGRAMMER'S AID sont inutilisables (car aux mêmes adresses). Maintenant, il y a en circulation des RAM de 32 K. Eh bien, elles ne vous permettront pas d'avoir plus d'octets pour BASIC. En effet, les 8 derniers K vont en $A000-$BFFF : ils sont donc inaccessibles à BASIC (sauf par POKE). De plus, ils rendent inutilisables les jeux, SUPER EXPANDER ou MASTER.

Adaptation des programmes haute résolution

Lorsqu'on a une extension d'au moins 8 K, la carte mémoire du VIC est bouleversée : les adresses mémoire d'écran et de RAM couleur sont modifiées. D'autre part les POKE dans le boîtier VIC pour obtenir la haute résolution ou les caractères programmables sont changés.

Ces questions ont déjà été traitées en partie dans La Commode. Nous rassemblons ici tous les éléments pour obtenir une solution, nous l'espérons, définitive.

1) Dans votre programme, toutes les données d'affichage doivent être paramétrées. Nous suggérons E (standard 7680) = origine RAM écran, CO (38400) = origine RAM couleurs, G (32768) origine générateur caractères.

2) Faire démarrer BASIC en $2000 (8192) par :

POKE 8192,0 : POKE 44,32 : NEW

(à taper en mode direct avant tout chargement ou frappe de votre programme).

3) Dans votre programme :

E devient 4096 ; CO devient 37888.

4) Sachant que, sans extension PEEK (36869) = 240, alors qu'il vaut 192 avec extension :

pour avoir G=	les POKE 36869 sont sans extension	avec extension
34816 (mode texte)	242	194
5120 ($1400) (haute	253	205
6144 ($1800) résolu-	254	206
7168 ($1C00) -tion)	255	207

Ceci montre que les POKE ont bien des équivalents en mémoire étendue contrairement à ce qui a été dit dans le N° 7.

Une autre méthode consiste à faire revenir le VIC aux répartitions d'adresses de la configuration non étendue, en faisant POKE 36869,240 ou ceux de la 2ème colonne du tableau. Simplement, il faut faire POKE 648,30 pour prévenir le système qu'on est revenu aux anciens paramètres.

Daniel-Jean DAVID

PRINT CHR$(8)

PRINT CHR$(8) a pour effet d'empêcher que la combinaison des touches SHIFT et COMMODORE fasse passer du mode graphique au mode texte.

Bien sûr PRINT CHR$(14) et PRINT CHRS$(142) agissent toujours ainsi que POKE 36869,242/240 (ou 194/192).

Cette inhibition cesse par PRINT CHR$(9).

Pierre Etienne THALBERG

POKE 646, ...

POKE 646, code couleur fait imprimer tout ce qui suit dans le code couleur indiqué. C'est donc équivalent à PRINT "CTRL couleur" à part que les codes = 8 (multicouleurs) sont accessibles.

La combinaison POKE 36878,0 : POKE 36879,24 : POKE 646,8 vous donne des caractères gras noirs sur fond blanc.

Pour les avoir en couleur de code C, faire :

POKE 36878,16*C : POKE 36879,24+C : POKE 646,C.

Pierre-Etienne THALBERG

ERRATUM (pas chez nous)

PROCEP vient de publier un intéressant catalogue de tous les produits VIC. Celui-ci est émaillé de quelques programmes. Celui de la page 7 (au demeurant fortement inspiré de "La Découverte du VIC" sans que crédit en soit donné) a une erreur. La ligne 510 doit se lire :

...POKE CH,PEEK(CH)OR(21(7-(XAND7)))...

Pierre-Etienne THALBERG

Routine GET

Une solution simple et pratique à un problème très fréquent, pour tous les systèmes du CBM 8096 au VIC 20.

Le jeu de Super Yam's présenté dans le n° 6 de La Commode utilise à plusieurs reprises la fonction GET. (lignes 321 à 323, 330 à 335, 351 à 363 etc...)

Le sous-programme ci-après permet d'éviter après chaque utilisation de GET, l'écriture de nombreuses lignes de programme destinées à tester le caractère saisi, et à effectuer les branchements nécessaires.

J'emploie personnellement cette routine dans tous mes programmes même si ce n'est que pour "pressez touche X pour action Y". Je la place personnellement en tête du programme, après les remarques de présentation.

Dans le cas du jeu "Super Yam's" elle pourrait être placée à partir de la ligne 70:

```
70   GOTO 140
80   REM ROUTINE SAISIE CARACTERE
82   POKE 158,0: LZ=LEN(Z$)
84   GET Y$: IF Y$="" THEN 84
86   FOR YY=1 TO LZ
88   IF MID$(Z$,YY,1)=Y$ THEN Y=YY
: YY=LZ: NEXT: RETURN
90   NEXT: POKE 158,0: GOTO 84
```

Les lignes 321 à 334 deviennent:

```
321  Z$="012345": GOSUB 80
322  ON Y GOTO
            350,330,330,330,330,340
330  L=Y-1: PRINT "! QUELS DES ?"
: Z$="12345": FOR E=1 TO L
331  GOSUB 80
334  F(E)=Y
```

Les lignes 351 à 363 deviennent:

```
351  Z$="LMD": GOSUB 80
352 ON Y GOTO 400,364,369
```

Les lignes 619 et 620 deviennent:

```
619  Z$="ON": GOSUB 80:
                ON Y GOTO 255,630
```

Les lignes 2902 à 2910 deviennent:

```
2902 Z$="ON": GOSUB 80:
                ON Y GOTO 3000,2910
2910 RETURN
```

Les transformations sont encore plus convaincantes lorsque le caractère saisi est alphanumérique, ou quand le nombre de caractères à tester est très grand, par exemple saisie par GET du choix d'un menu composé de nombreuses lettres de l'alphabet.

Que fait donc cette routine ?

La ligne 82 vide le tampon du clavier (POKE 198,0 pour le VIC) et donne à la variable LZ la valeur de la longueur de Z$. Z$ est une chaîne composée de tous les caractères autorisés.

La ligne 84 attend la saisie d'un caractère dans Y$.

Les lignes 86 à 90 testent si le caractère saisi est un des caractères contenus dans la chaîne Z$.

Si le caractère est reconnu, la variable Y prend la valeur du rang de Y$ dans Z$, la boucle est close et on sort du sous-programme. Y pourra alors servir dans un branchement du type ON Y GOTO ou ON Y GOSUB.

Si le caractère n'est pas reconnu, on recommence, après avoir vidé le tampon clavier.

On peut insérer dans la ligne 90, entre POKE 158,0 et GOTO 80, le renvoi à un sous-programme d'erreur ou de déclenchement d'un BEEP.

Pour les programmeurs qui auraient des problèmes de mémoire, cette routine peut s'écrire en une ligne:

```
70   GOTO 140
80   POKE 158,0: GET Y$: FOR Y=1 TO
  LEN(Y$): IF Y$<>MID$(Y$,Y,1) THEN
  NEXT Y: GOTO 80
82   RETURN
```

L'inconvénient de cette dernière solution est que l'on sort de la boucle sans libérer la pile, et qu'on ne peut le faire qu'un nombre limité de fois avant d'arriver au message d'erreur "OUT OF MEMORY".

Conclusion

L'utilisation régulière de cette routine la rend très vite indispensable. Elle permet de faire des saisies de caractères très élaborées, mais reste néanmoins compréhensible par tous, tout en permettant une maintenance aisée et rapide.

Attention cependant de ne pas utiliser dans votre programme une des variables de la routine.

Martial MARTINEZ

Complétez votre Collection

Les numéros anciens de La Commode sont encore disponibles mais peut-être pas pour longtemps. Hâtez vous de compléter votre collection si vous ne voulez pas rater des renseignements essentiels.

BON DE COMMANDE

à envoyer à La Commode 28, rue Vicq-d'Azir 75010 PARIS

Je désire les numéros suivants de La Commode (entourer)

1 2 6 7

Joignez votre réglement (35F par numéro) à l'ordre de SEDERMI

☐ CCP ☐ CB ☐ MANDAT

Bientôt épuisés

NOM .....

ADRESSE .....

.....

.....

Les Ficelles

Quoi ? Encore une rubrique d'astuces, à côté de VIC à BRAC et de PETits trucs et CoMBines ?

-Oui. Il s'agit ici des astuces applicables à la fois au VIC et au PET/CBM. Formidable, non ?

Interruption d'un listing

Aucun listing d'un programme qui se respecte ne tient en entier sur l'écran, à fortiori sur le VIC avec ses micro-lignes. Quant à la touche RVS ou flèche à gauche ou CTRL, elle ne ralentit pas assez pour qu'on puisse examiner ce qu'on veut.

Notons que sur certains modèles, avec certains logiciels, on a une fonction d'arrêt des écritures sur écran et reprise (Ex. sortie d'un catalogue avec DOS SUPPORT et bien d'autres).

Ah, si nous pouvions faire que, en appuyant sur SHIFT, tout s'arrête et reprenne lorsqu'on relâche SHIFT ! Nous choisissons la touche SHIFT pour cette fonction parce qu'avec la touche SHIFT LOCK chère au Club Magenta, on peut avoir les deux mains libres pendant l'arrêt : on peut même aller chercher un bout de papier ou un livre pendant l'arrêt.

Eh bien, c'est parfaitement possible grâce à l'interruption périodique du système. Vous savez sans doute que tous les 1/60 ème de seconde, le processeur est interrompu de son travail normal (par exemple exécuter une instruction BASIC) pour exécuter une routine dite d'interruption. En particulier, au cours de cette routine, le système met à jour l'horloge temps réel et examine le clavier. Ensuite, le processeur revient au programme interrompu.

Il se trouve que l'adresse où le système doit se rendre lorsqu'il est interrompu est fournie de façon indirecte en RAM (aux adresses $314, $315 pour un VIC - on donnera plus loin les équivalents PET/CBM). Il y a à ces adresses BF et EA ce qui forme l'adresse $EABF qui est l'adresse normale d'interruption du VIC. Or, on peut changer le contenu du vecteur d'interruption puisqu'il est en RAM pour y mettre l'adresse d'une routine utilisateur. Et alors, il est garanti que 60 fois par seconde, le système ira exécuter cette routine.

Si nous prenons comme routine utilisateur, la suivante :

examiner le clavier
touche SHIFT enfoncée ?
si oui boucler sur examen du clavier
si non, aller à la routine normale d'interruption.

nous avons la solution de notre problème : lorsque la touche SHIFT est enfoncée, le système s'arrête dans notre routine d'interruption, donc l'écran reste figé pour être examiné à loisir jusqu'à ce qu'on lâche la touche.

Voici la routine écrite en langage machine et implantée dans le tampon cassette. Elle est en deux parties. Une première partie établit le vecteur d'interruption pour qu'il pointe vers la routine utilisateur. La 2ème partie est la routine utilisateur proprement dite.

Le lancement se fait par SYS 828.

Nous donnons ci-dessous la version PET/CBM grâce aux corres-

pondances :

	VEC	INTNOR
CBM	$0090	$E62E
PET	$0219	$E685
BASIC4	$0090	$E455

Mais le test de la touche SHIFT doit être différent car le balayage du clavier n'est pas en sous-programme (le système du VIC est mieux écrit). Aussi, sur CBM 4000 ou 8000, la routine devra être placée ailleurs si l'on utilise les disques en BASIC 4.

Dans tous les cas, avant d'utiliser les cassettes ou l'IEEE, il faut rétablir le vecteur d'interruption. Sur VIC, cela se fait par STOP/RESTORE. Sur PET/CBM, il faut un sous-programme analogue à la première partie qui rétablisse les adresses.

Il faut noter que sur VIC, la touche SHIFT est difficile à utiliser lorsque cette routine est activée. Sur CBM, elle est impossible, et les touches 0,), <, etc. (touches de la même ligne de balayage sont très difficiles. Ce programme est fait pour figer les sorties, pas pour faciliter les entrées.

Pierre Etienne THALBERG

Pour écrire sur deux périphériques à la fois

Ex. : un fichier disque et une imprimante (pourrait être deux imprimantes).

```
OPEN 8,8,8,"0:LLL,S,W" : OPEN4,4
CMD8 : CMD4 : LIST
```

pour terminer

```
CMD8 : PRINT#4: CLOSE8 : CLOSE4
```

Bien sûr, au lieu de LIST, vous pouvez employer PRINT (pas PRINT#). Cela va plus vite que d'employer deux PRINT# : PRINT#8,... : PRINT#4,... mais, bien sûr, on va à la vitesse du périphérique le plus lent.

Cela ne marche que pour deux périphériques IEEE, donc pas pour l'écran + 1 imprimante.

Sur VIC, cette opération marche, mais à condition d'inverser l'ordre : mettre

```
CMD4 : CMD8 : LIST
```

et

```
CMD4 : PRINT#8 : ...
```

Daniel Jean DAVID

```
                    ; VERSION VIC
                    ;
828  033C 78              SEI          ; première partie-masque interruption
     033D A9 49           LDA #$49     ; partie basse de l'adresse
     033F 8D 14 03        STA VEC
     0342 A9 03           LDA #$03     ; partie haute
     0344 8D 15 03        STA VEC+1
     0347 58              CLI          ; rétablit interruption
     0348 60              RTS
     0349 20 9F FF P      JSR CLA      ; deuxième partie-examine clavier
     034C 4E 8D 02        LSR SHIFT    ; indicateur de SHIFT dans retenue
     034F B0 F8           BCS P        ; boucle si SHIFT
     0351 4C BF EA        JMP INTNOR   ; sinon va à inter. normale

                    ; VERSION PET-CBM (assemblée pour 3000)
                    ;
828  033C 78              SEI          ; première partie-masque interruption
     033D A9 47           LDA #$47     ; $45 sur PET
     033F 85 90           STA VEC      ; pas en page 0 sur PET
     0341 A9 03           LDA #$03     ; partie haute
     0343 85 91           STA VEC+1
     0345 58              CLI          ; rétablit interruption
     0346 60              RTS
     0347 A9 08     PS    LDA #$08     ; deuxième partie-ligne de balayage de
     0349 8D 10 E8        STA $E810    ; SHIFT dans PIA clavier
     034C EE 11 E8        INC $E811    ; si pas de SHIFT : $E811 =$FF
     034F D0 F6           BNE PS       ; donc résultat nul
     0351 4C 2E E6        JMP INTNOR   ; sinon va à inter. normale
```

Définition d'O.G.D. multicouleurs

Tant le VIC que le 64 disposent de modes d'affichage dits "multicouleurs" qui ont été jusqu'ici bien peu explorés. Une des utilisations les plus intéressantes de ces modes est celle qui permet de définir, sur 64, des O.G.D. ("sprites") multicouleurs.
(N.B. Cet article est en partie extrait de "La Découverte du 64" ed. du P.S.I., parution au SICOB 83.)

Les modes multicouleurs

Que l'on soit dans le cas du VIC ou du 64, l'affichage à la résolution la plus haute (VIC: 184x-176; 64: 200x320) se fait avec un choix restreint de couleurs. L'état d'un point élémentaire est défini par un bit (0="éteint", 1="allumé"). Ici, "éteint" signifie "de la couleur du fond de l'écran", "allumé" signifie "de la couleur spécifiée dans la mémoire des couleurs".

Mais, et c'est là que se manifeste la contrainte, la mémoire de couleur n'est définie qu'à l'échelle de la maille caractère (VIC: 23x22; 64: 25x40). Donc, si deux points allumés peuvent être de couleurs différentes s'ils sont dans des mailles différentes, ils sont obligatoirement de la même couleur s'ils sont dans la même maille. Ceci revient à diviser par 8 (dans chaque direction) la résolution de la couleur par rapport à la résolution géométrique.

Ceci est suffisant dans les applications de tracé de courbes (on trace la courbe d'une couleur et le fond est d'une deuxième couleur) mais certaines applications peuvent ne pas s'en contenter. C'est pourquoi le VIC et le 64 proposent des modes multicouleurs.

Leurs principes fondamentaux sont toujours les mêmes:

1- on sacrifie un facteur 2 en résolution horizontale. Maintenant les deux points N°s 2p et 2p+1 auront le même destin défini par la paire de bits correspondants.

2- Cette paire de bits forme un code de couleur indirect : 2 bits ne donneraient qu'un choix de 4 couleurs possibles. Avec l'indirection les deux bits spécifient l'endroit où est à prendre le code couleur complet.

Avant d'appliquer ceci aux OGD (Objets Graphiques Déplaçables) ou "sprites", nous devons voir les principes de la définition des OGD.

Généralités sur les OGD

On appelle OGD un dessin défini dans une matrice 21 sur 24 déplaçable à volonté sur l'écran.

Le 64 peut gérer 8 OGD simultanément et ce, indépendamment de tout le reste de l'affichage. Les OGD sont numérotés de 0 à 7. Il faut noter que si un OGD apparaît sur l'écran au même endroit qu'un autre affichage (p.ex. un caractère) il a priorité, c'est-à-dire qu'il occulte cet autre objet. De même, si deux OGD se rencontrent sur l'écran, celui de n^{o} le plus petit a priorité, donc il occulte l'autre. D'autre part, une rencontre sur l'écran entre un OGD et un autre affichage ou un autre OGD s'appelle une collision : elle est détectée par le système et c'est très utile pour la programmation des jeux.

Les OGD sont définis par des

emplacements mémoire par des registres du boîtier 6567 propres à chaque OGD ou par des registres partagés entre les OGD. Lorsqu un registre est partagé le bit n° i concerne l OGD n° i.
Un OGD est défini par :

1 - sa forme. Celle-ci se définit de la même manière qu un caractère programmable mais dans une matrice 21 lignes 24 colonnes arrangées ainsi :

```
col     12...78910...15161718...2324
bits    76...107 6... 1 0 7 6... 1 0

lig. 1 octet 0  octet 1  octet 2
lig. 2 octet 3  octet 4  octet 5
lig. 3 octet 6  ..............
          .          .
          .          .
          .          .
lig.21 octet 60 octet 61  octet 62
```

Il faut donc 63 octets pour définir chaque OGD. En fait, on accorde un 64ie octet de remplissage car 64 est une puissance de 2. Cela simplifie l'adressage. (dans la suite, on numérotera les lignes de 0 à 20 et les colonnes de 0 à 23).

Les adresses sont définies de manière indirecte à l aide de pointeurs en 2040 à 2047 (derrière la mémoire d écran).

Le pointeur d adresse 2040 + I concerne l OGD n° I (I de 0 à 7). L adresse de l octet 0 de l OGD n° I est 64 * PEEK (2040 + I).

L adresse de l octet K de l OGD n° I est 64*PEEK (2040 + I) + K . On y met le motif binaire qui définit la zone correspondante de l OGD.

Notez que cette définition indirecte de l adresse permet de définir plusieurs OGD de même forme: il suffit que plusieurs pointeurs soient égaux; ils pointeront vers la même adresse mémoire et donc définiront la même forme. Cela permet aussi de tenir en réserve plusieurs formes pour le même OGD : on passe très rapidement d'une forme à l autre par un seul POKE dans le pointeur.

2 - Son activation :

L OGD n° I apparaît si le bit I de l octet 53269 est à 1. Il disparaît si ce bit est à 0.

Ex. faire apparaître l'OGD n° I :
POKE 53269, PEEK (53269) OR 2I

3 - Son mode multicouleurs :

L'OGD n° I (activé par le POKE en 53269) est mis en mode multicouleurs si l'on met à 1 le bit n° I de l'octet d'adresse 53276 :
POKE 53276, PEEK (53276) OR 2I.

4 - Sa couleur :

Pour définir la couleur de l'OGD n° I, on fait :
POKE 53287 + I, code couleur.

5 - Sa dilatation :

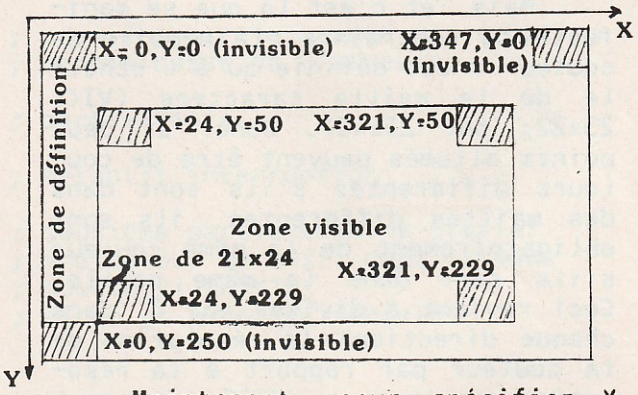
On dilate en horizontale l'OGDn° I par :
POKE 53277, PEEK (53277) OR 2I

On le dilate en verticale par
POKE 53271, PEEK (53271)OR 2I

6 - Sa position sur l'écran :

C'est la position du coin supérieur gauche qu on définit, par X compris entre 0 et 347 et y compris entre 0 et 255.

A noter que l OGD n est visible sur l'écran que pour X compris entre 24 et 321 et Y compris entre 50 et 229. En dehors l OGD disparaît dans les bords, ou même hors de l'écran, voir fig. suivante.



Maintenant, pour spécifier Y c'est facile. On fait POKE 53249 + 2 * I, Y (I = numéro de l OGD)

Pour X, c'est plus difficile car X ne tient pas en un octet. Si X 255 il y a un 9ème bit à 1. Pour l'octet bas de X, il y a un registre pour chaque OGD:

POKE 53248 + 2 * I, X AND 255

Les 9èmes bits des X des 8 OGD sont, eux, regroupés dans le registre 53264 :

POKE 53264 (PEEK (53264) AND (255-2I)) OR (-2I) * (X 255)

Cette expression a l'air compliqué, mais elle ne fait qu'exploiter le fait qu'une expression logique est évaluée comme 0 si elle est fausse et -1 si elle est vraie.

Revenons sur l'étape 1. Il nous faut choisir les adresses mémoire où les OGD seront définis, donc quelles valeurs mettre en 2040 à 2047.

Il y a 2 choix conseillés :

a) si vous n'avez que 3 OGD au plus, vous pouvez faire :

POKE 2040, 13 (adresses de l'OGD 0: 832-895)

POKE 2041, 14 (adresses de l'OGD 1: 896-959)

POKE 2042, 15 (adresses de l'OGD 2: 960-1023)

.... ce qui revient à placer les OGD dans le tampon cassette, donc est valable pendant que vous n'utilisez pas la cassette.

b) on peut mettre les données en 12288 et suivantes ce qui se fait par :

POKE 2040 + I, 192 + I

Ne pas oublier de protéger de BASIC la zone 12288 par :

POKE 52, 48 : POKE 56, 48 : CLR

Le programme listé à la fin de l'article est un "éditeur d'OGD". Il vous permet de définir visuellement vos OGD dans une grille de points sur l'écran : les . seront des vides tandis que les X représenteront les éléments allumés de l'OGD. Pour allumer un X à son emplacement, vous y allez par mouvements de curseur sur la ligne. Vous terminez la ligne par RETURN.

Le programme se termine par la visualisation de l'OGD qu'on vient de créer pendant 5 s, puis il reprend la création d'un nouvel OGD (vous arrêtez en disant que le n° de l'OGD à créer est =8; vous pouvez recommencer la création du même OGD si vous n'êtes pas satisfait en redonnant le même n°)

Le programme est aux N°s de ligne 60010 et suivantes pour pouvoir être incorporé à la fin de votre propre programme d'utilisation : vous faites d'abord RUN 60000 pour créer vos OGD, puis RUN pour les utiliser .

Le programme vous permet, en fait, d'éditer vos OGD, c'est-à-dire de les modifier à partir d'un dessin préalable gardé sur disque ou sur cassette. Pour cela, vous répondez "OUI" à la question "déjà défini?". On vous demande ensuite le nom du fichier et le n° du périphérique. Vous répondez le nom choisi et 1 pour la cassette, 8 pour le disque. Si, comme nom, vous répondez "$", le motif précédent est cherché en mémoire : c'est que vous souhaitez modifier l'OGD que vous venez juste de définir. En fin de visualisation, on vous demande si vous voulez sauver votre dessin.

En 60240, on vous demande de quelle couleur vous voulez votre OGD et on assure qu'il soit monochrome. Les instructions 60280 à 60320 traitent le cas multicouleurs que nous examinons maintenant.

OGD multicouleurs

Dans ce que nous avons vu jusqu'à maintenant, chacun des 8 OGD possibles peut avoir sa propre couleur, mais il ne peut en avoir qu'une (à part des "trous" de la couleur du fond de l'écran).

En mode multicouleurs, chaque OGD pourra avoir trois couleurs en plus de celle de l'écran. Conformément au principe déjà explicité, cela se paye par une résolution horizontale divisée par 2 : l'OGD est maintenant formée de 21 lignes de 12 taches élémentaires. Chaque tache est représentée par un couple de bits dans les mémoires de définition. Vis à vis de l'éditeur d'OGD tout se passe de la même façon mais chaque tâche est définie par 2 caractères . ou X. Les motifs possibles sont :

00 (..) couleur du fond

01 (.X) couleur dont le code est dans le registre 52285 (couleur C1 dans le programme éditeur)

10 (X.) couleur de l'OGD (registre 53287 + I) (c'est la couleur unique de l'OGD en mode mono chrome)

11 (XX) couleur dont le code est dans le registre 53286 (couleur C2 dans le programme éditeur).

L'éditeur vous permet de définir vos OGD en multicouleurs et il vous permet une seconde visualisation pour juger de l'effet obtenu.

Limitation

Une limitation apparaît aussitôt : si la couleur '10' est propre à chaque OGD, les couleurs '01' et '11' sont, elles, les mêmes pour tous les OGD. Les multicouleurs devront être choisies avec soin. Le programme "FINISH" (course d'autos) de ce numéro en est un bon exemple : chaque voiture est un OGD, les voitures ne diffèrent entre elles que par la couleur de la cabine.

Point qui apporte de la souplesse, au contraire, chaque OGD peut au choix être en mode monochrome ou multicouleurs.

Conclusions

1 - Les OGD sont un des domaines où le mode multicouleurs a le plus d'intérêt. Au contraire, ce mode n'a guère d'utilité en graphiques haute résolution et encore moins en mode caractères, encore que vous pouvez voir dans VIC à BRAC une application du mode multicouleurs pour obtenir des caractères gras. Il va sans dire que la même technique est adaptable au 64.

2 - Nous n'avons pas abordé, dans cet article, tout ce qui concerne les OGD (il faudrait des volumes). Toutefois, on a vu certains aspects importants et l'éditeur d'OGD devrait vous aider à expérimenter. Rappelons que dans le n° 7, il y avait un autre éditeur d'OGD, sous une autre présentation. Examinez-le aussi.

Daniel-Jean DAVID

EDITEUR D'OGD

```
60000 POKE52,48:POKE56,48:CLR
60010 DIM A$(20):CR$=CHR$(13)
60020 FOR I=0 TO 7:POKE 2040+I,192+I:NEXT
60030 INPUT "TWO DE L'OGD A DEFINIR";I:IF I>7 THEN END
60040 A=64*PEEK(2040+I)
60050 FOR M=A TO A+63:POKE M,0:NEXT
60060 INPUT"DEJA DEFINI ";R$
60070 IF R$="OUI" GOTO 60100
60080 FOR L=0 TO 20
60090 A$(L)="....................":NEXT :GOTO 60150
60100 INPUT"NOM FICHIER,PERIPH.";N$,P:AS=-2*(P=8)
60110 IF N$="$" GOTO 60150
60120 OPEN 1,P,AS,N$
60130 FOR L=0 TO 20:INPUT#1,A$(L):NEXT
60140 CLOSE1
60150 FOR L=0 TO 20
60160 PRINT"? "A$(L):NEXT
60170 PRINT"TTTTTTTTTTTTTTTTTTTTT";
60180 FOR L=0 TO 20:INPUT A$(L)
60190 FOR K=0 TO 23
60200 J=INT(K/8):B=K AND 7
60210 B$=MID$(A$(L),K+1,1):M=A+3*L+J:C=PEEK(M)
60220 IF B$="X" THEN POKE M,C OR 2↑(7-B)
60230 NEXT:NEXT
60240 INPUT"TCODE COULEUR";CS :POKE53276,PEEK(53276) AND (255-2↑I)
60250 POKE 53287+I,CS:POKE 53269,PEEK(53269) OR 2↑I
60260 POKE 53249+2*I,100:POKE 53248+2*I,100:POKE53264,0
60270 FOR T=1 TO 5000:NEXT
60280 INPUT"TMULTICOULEURS";R$
60290 IF R$<>"OUI" GOTO 60330
60300 INPUT"CODES COULEURS";C1,C2
60310 POKE 53285,C1:POKE53286,C2:POKE53276,PEEK(53276)OR2↑I
60320 FOR T=1 TO 5000:NEXT
60330 INPUT"ON SAUVE";R$:IF R$<>"OUI" GOTO 60380
60340 INPUT"NOM FICHIER,PERIPH.";N$,P:AS=1-2*(P=8)
60350 OPEN 1,P,AS,N$+",S,W"
60360 FOR L=0 TO 20:PRINT#1,A$(L);CR$;:NEXT
60370 CLOSE1
60380 POKE 53269,PEEK(53269) AND (255-2↑I)
60390 GOTO 60030
```

Atari - Trucs

Gestion des erreurs

Avec l'ATARI, le programme peut garder le contrôle en cas d'erreur. Il suffit, préalablement à l'erreur, donc en début de programme, d'exécuter une instruction :

TRAP numéro.

En cas d'erreur, vous vous retrouverez en 'numéro'. La première chose à faire à cette ligne est de savoir quelle est l'erreur. C'est facile : PEEK(195) est le n° de l'erreur (la liste est dans le manuel ATARI ou dans "La Découverte de l'ATARI") :

```
Ex. : 10    TRAP 2000
      50    READ A
      2000  IF PEEK (195)=6 THEN
            RESTORE:GOTO 50
      2010  PRINT"ERREUR NO";
            PEEK(195):END.
```

On voit le mécanisme: on est dans une série de DATA en 50. Quand on arrive à la fin, il se produit l'erreur 6. On fait RESTORE et on retourne lire. Si l'erreur n'est pas 6, on affiche le n° et on arrête.

Adresse de la mémoire d'écran

2 méthodes :

```
1) D=PEEK(560)+256*PEEK(561)

   E=PEEK(D+4)+256*PEEK(D+5)

2) E=PEEK(88)+256*PEEK(89)
```

Fusion des programmes

Alors que c'est toute une histoire avec les Commodore et l'Oric, il est très facile de fusionner des programmes ou des sous-programmes sur l'ATARI.

En effet, la commande LIST peut s'adresser à tout autre périphérique que l'écran. Pour la casette faire :

LIST"C:" 1000,2000

Ceci envoie le sous-programme compris entre 1000 et 2000 sur la casette.

Un tel fichier ne peut pas être relu par LOAD ni par CLOAD. Mais il peut l'être par ENTER :

ENTER"C:"

Mais ENTER ne supprime pas le programme déjà présent en mémoire : les nouvelles lignes lues vont s'intercaler à la place voulue par leurs numéros ; c'est exactement une fusion. Sur Commodore, il faut faire appel à EDEX, Programmer's Aid ou Librarian.

Daniel-Jean DAVID

Suppression de lignes en bloc

```
20000 INPUT "DEBUT,FIN";D,F
20010 FOR L=D TO F
20020 PRINT CHR$(125):PRINT:PRINT
20030 PRINT L
20040 PRINT"CONT":POSITION 0,0:
      POKE 842,13:STOP
20050 POKE 842,12:NEXT L
20060 PRINT CHR$(125):END
```

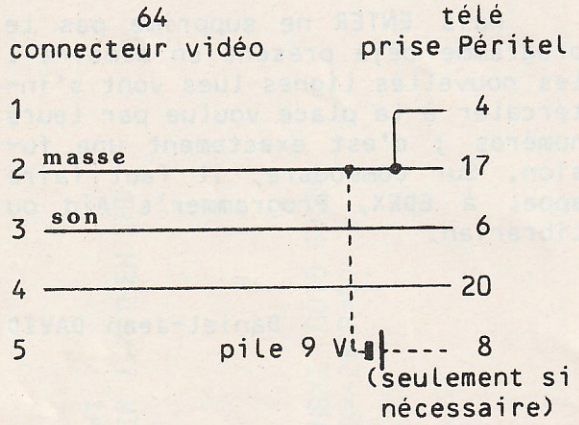
Quand vous avez une série de lignes à supprimer faites : GOTO 20000. (Il faut que les n°s à supprimer soient inférieurs à 20000, sinon, changez les numéros des lignes du programme de suppression.

Pierre-Etienne THALBERG

64 à Brac

Cordon Péritel pour le 64

Attention, le cordon Péritel du VIC (voir La Commode n° 4) ne marche pas sur le 64. Voici le branchement à effectuer :



Si ce branchement ne suffit pas c'est qu'il faut alimenter en + 12 V la broche 8 de la prise Péritel. En pratique une pile de 9 V fixée à la prise péritel (- à la masse 4, + au 8) suffit.

Pierre-Etienne THALBERG

Désactivation de l'écran

POKE 53265,11 vide l'écran. En fait, cela désactive le 6567, ce qui évite les vols de cycles au processeur. Cela a plusieurs applications :

- extinction de l'écran pour dessins animés,
- petite accélération des calculs,
- possibilité d'utiliser les périphériques : le logiciel de gestion cassette le fait automatiquement, mais l'extinction de l'écran est par ailleurs désagréable.

Il faudrait le faire pour utiliser le RS 232.

Enfin, ce POKE vous permet d'utiliser l'unité de disque 1540 du VIC sans changement de ROM.

Vous rétablissez l'affichage par POKE 53265,27 (à taper en aveugle) ou STOP RESTORE.

Erreurs de TAB et SPC corrigées

Il semble que TAB et SPC fonctionnent correctement sur le 64, alors que sur VIC, TAB et SPC donnaient une erreur de syntaxe s'ils étaient le premier élément d'une liste d'impression.

Pierre-Etienne THALBERG

Premières notions sur l'architecture du 64

Pour utiliser plus efficacement votre 64

Les Boîtiers

Le 64 est bâti autour d'un microprocesseur 6510 qui n'est rien d'autre qu'un 6502 ayant en plus un port d'entrées/sorties parallèles dont nous verrons l'usage dans cet article. Sinon, son espace adressable reste de 64 K ce qui pose les plus gros problèmes d'architecture dont nous allons parler.

Côté mémoires, il y a d'abord les 64 K octets de RAM dynamique (8 boîtiers de 64 K x 1 bit), deux boîtiers de 8 K octets chacun de ROM pour le système d'exploitation et BASIC. On peut, à l'extérieur connecter d'autres ROM par le connecteur d'extension. Enfin, le générateur de caractères réside dans une ROM de 4 K (2 K x 2 jeux de caractères), qui est, ou n'est pas incluse dans l'espace adressable selon les circonstances.

Du côté des boîtiers d'entrées/sorties, on trouve d'abord deux 6526 (2 ports parallèles chacun - ils ne sont pas très différents des 6520 ou 6522 que vous connaissez probablement). Ensuite, le synthétiseur sonore 6581 et le V.I.C. (Vidéo Interface Chip = boîtier d'interface vidéo). Tous ces boîtiers sont vus du processeur comme des cases mémoire. Le V.I.C. est le véritable maître du système, plus que le microprocesseur : c'est lui qui fournit l'horloge au système, les signaux RAS et CAS de gestion de la RAM dynamique etc... En outre il accède à la mémoire, mais seulement à un banc de 16 K octets.

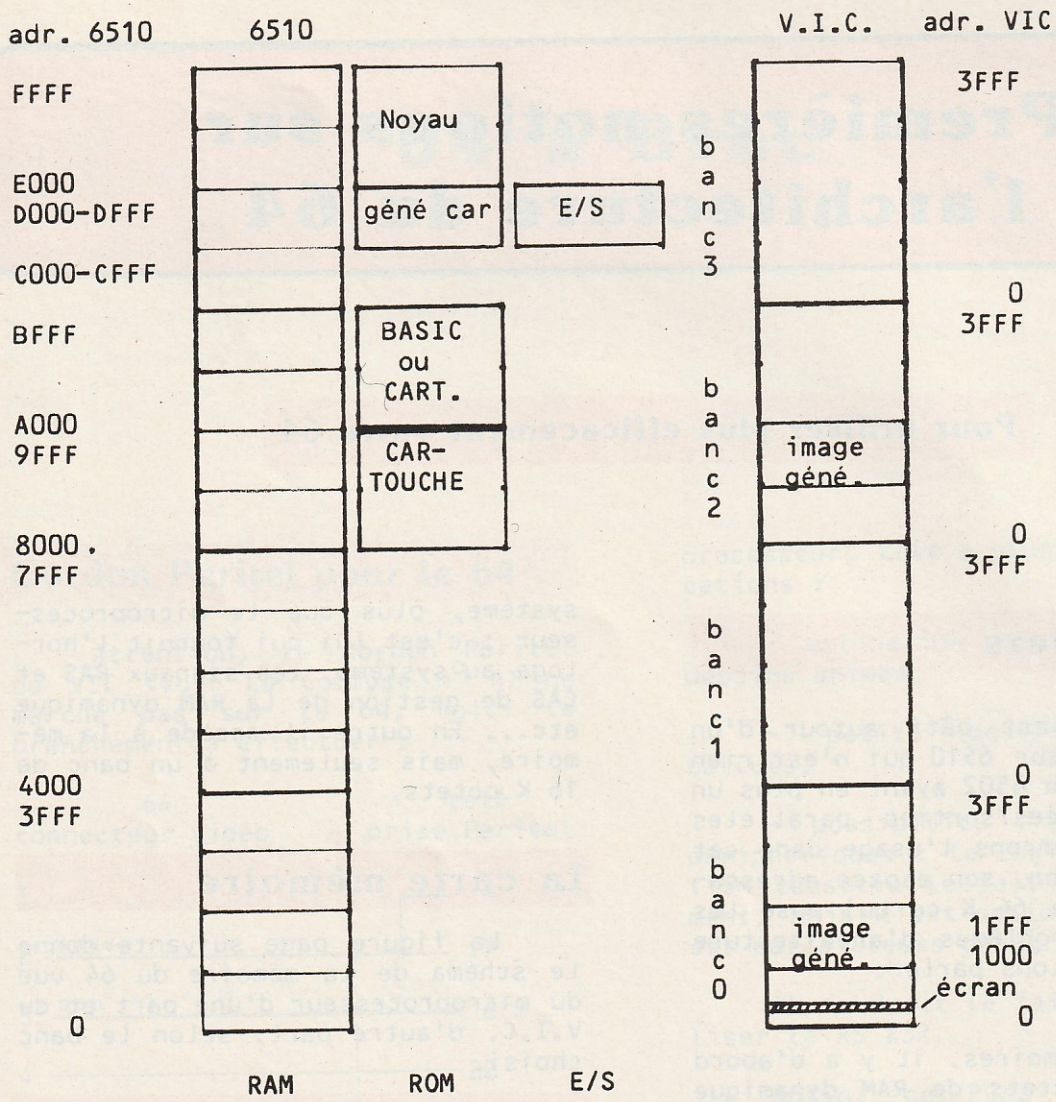
La carte mémoire

La figure page suivante donne le schéma de la mémoire du 64 vue du microprocesseur d'une part et du V.I.C. d'autre part, selon le banc choisi.

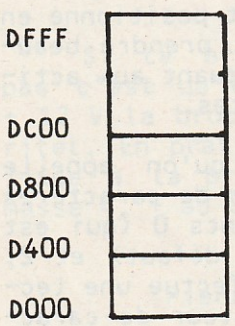
Le V.I.C.

La plupart des besoins en mémoire du V.I.C. sont en RAM, donc lorsque le V.I.C. est positionné en banc 2 ou 3 il faut prendre beaucoup de précautions quant aux activations qui sont faites.

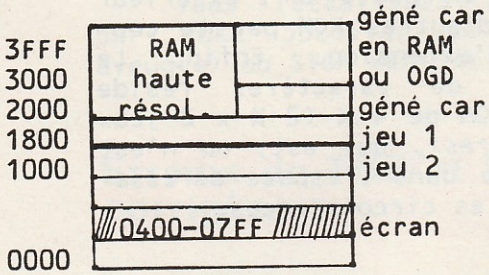
Clarifions ce qu'on appelle l'image du générateur de caractères en ROM. Dans les bancs 0 (qui est le banc assigné par défaut) et 2, lorsque le V.I.C. effectue une lecture dans son générateur de caractères entre 1000-1FFF, il n'accède pas à la mémoire correspondante (1000-1FFF ou 9000-9FFF) mais à la ROM du générateur de caractères qui a -pour le V.I.C. uniquement- une image à ces adresses. Au contraire, ce phénomène ne se produit pas dans les bancs 1 et 3 : donc, dans ces bancs, les jeux de caractères standard seront inaccessibles.



Détail du bloc des E/S



Détail d'un bloc du V.I.C.
avec les assignations par défaut



Les signaux d'activation

Il y a 4 zones litigieuses dans la mémoire, qui sont partagées entre deux éléments ou plus :

de 8000 à 9FFF : RAM ou cartouche d'extension ROM

de A000 à BFFF : RAM ou ROM (BASIC ou cartouche)

de D000 à DFFF : RAM ou E/S ou générateur de caractères

de E000 à FFFF : RAM ou ROM (Noyau)

Les choix sont commandés par les signaux suivants :

B (bit 0 du port d'adresse 0001) choix RAM(0)/ROM(1) en A000-BFFF

K (bit 1 du port d'adresse 0001) choix RAM(0)/ROM(1) en E000-FFFF

G (bit 2 du port d'adresse 0001) active le générateur de caractères au lieu des E/S en D000-DFFF

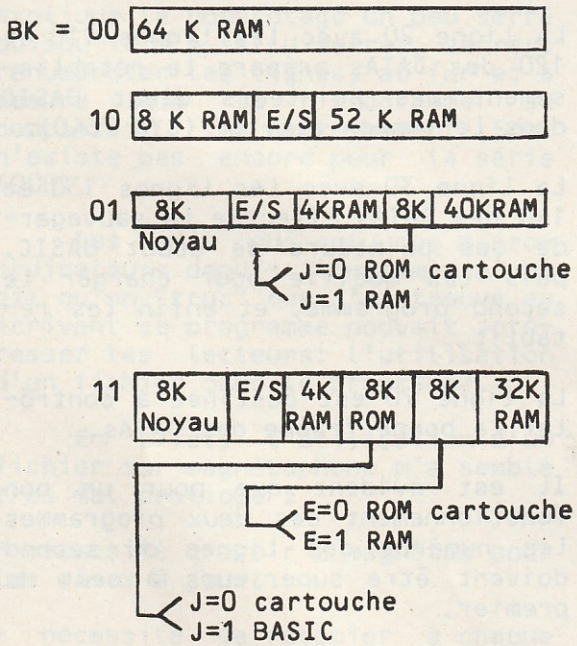
E (fourni par une cartouche) active la ROM d'extension au lieu de la RAM en 8000-

J (fourni par une cartouche) active la ROM d'extension au lieu de BASIC en A000-

Ces deux derniers signaux sont sur le connecteur d'extension. Ils sont tirés à 1 en l'absence de cartouche, mais une cartouche d'extension peut les mettre à la masse.

La combinaison BK = 00 active la RAM en D000-DFFF. Sinon on a en général les Entrées/Sorties actives dans cette zone (sauf si G = 0 qui active la ROM caractères).

Si l'on élimine la combinaison E = 1, s = 0 qui donne une configuration de machine de jeux, on a l'arborescence :



Dans toutes les configurations où les E/S sont actives, si l'on met G à 0 (par POKE 1, PEEK (1) AND 251), le générateur de caractères en ROM devient accessible au microprocesseur. Cela ne sert que pour copier en partie ce générateur en RAM dans une application graphique. Ensuite, il faut réactiver les entrées/sorties par POKE 1, PEEK(1) OR 4, sinon le 64 ne peut pas faire grand-chose.

Dans les configurations à 64 K RAM, il faut :

- être exclusivement en langage machine (BASIC n'est pas là),

- écrire ses propres routines d'entrées/sorties (le Noyau du système d'exploitation est innaccessible).

- activer les E/S de temps en temps par LDA$1 ; ORA 1 ; STA$1 (à chaque fois qu'on a besoin de faire une E/S) et les désactiver lorsque l'opération est faite.

Les configurations les plus utilisées sont :

B=K=J=E=1 (configuration à la mise sous tension)
avec BASIC, le noyau et 40K de RAM,

B=K=J=1 E=0
avec BASIC, le noyau, 32K de RAM et une ROM d'extension

B=0 K=J=1 E=0 ou 1
avec le noyau et 52K de RAM pour charger un langage ou CP/M

Conclusion

Maintenant que nous avons compris comment le 64 organise ses différentes configurations, nous verrons dans le prochain article de cette série comment les exploiter en relation avec les bancs 1,2 et 3 de l'espace mémoire du V.I.C. pour que les programmes graphiques ne restreignent pas trop la place mémoire restant à BASIC.

Pierre-Etienne THALBERG

Chaînage de 2 programmes

Un "Utilitaire" bien utile !

```
10FOR T=0 TO 8: READ A: POKE
7424+T,A: NEXT
20FOR T=0 TO 30: READ A: POKE
7584+T,A: NEXT
30FOR T=0 TO 33: READ A: POKE
7616+T,A: NEXT
60SYS 7424
70RESTORE: FOR T=0 TO 73: READ A :
B=B+A: NEXT: PRINT B: IF B<>7768
GOTO 90
80FOR T=1 TO 500: NEXT: NEW
90PRINT "(CLR)(DOWN)(DOWN)ERREUR
DANS LES DATAS": STOP
100DATA 160,29,162,0,132,56,134
,55,96
110DATA 160,0,185,182,29,153,119
,2,200,192,8,208,245,185,182
120DATA29,133,198,76,192,29,0,83,89
,83,55,54,52,48,13,8
130DATA166,44,164,43,142,255,3
,140,254,3,164,45,166,46,136,136
140DATA16,1,202,134,44,132,43,96,17
4,255,3,172,254,3,76,211,29,0
```

Analyse du programme

1-Charger et exécuter ce programme: s'il n'y a pas d'erreur dans les "DATAs" il s'efface de lui-même.

2-Charger normalement le premier programme.

3-Pour charger le deuxième programme faire:
SYS 7584: LOAD "NOM"

4-En cas d'utilisation de la touche "STOP" pendant le chargement rétablir les pointeurs de début par SYS 7640.

5-En cas d'utilisation du bouton "RESET" pour ceux qui l'on monté, réinitialiser le sommet mémoire par SYS 7424.

N.B.: ce programme est destiné à un VIC 20 + 3K de MEV.

Utilisation

La ligne 10 avec la ligne 100 des DATAs est destinée à mettre le sommet de la mémoire à 7424 pour protéger l'utilitaire. Ces 256 octets réservés sont vus larges afin d'adjoindre d'autres utilitaires machine dans cet espace. Cette protection sera effective lors de l'exécution de la ligne 60.

La ligne 20 avec les lignes 110 et 120 des DATAs prépare le rétablissement des pointeurs début BASIC dans le tampon clavier (SYS 7640).

La ligne 30 avec les lignes 130 et 140 des DATAs effectue la sauvegarde des pointeurs de début BASIC, puis les modifie pour charger le second programme, et enfin les rétablit.

La ligne 70 est destinée à contrôler la bonne frappe des DATAs.

Il est évident que pour un bon fonctionnement des deux programmes les numéros des lignes du second doivent être supérieurs à ceux du premier.

Pierre ROBERT

Inflation

Comme l'inflation ne s'arrêtera pas de sitôt,ce programme vous servira longtemps : il est prévu pour être mis à jour jusqu'en 1999.

Ce programme utilise un fichier inclus dans le programme avec mise à jour du fichier par création de lignes de programme par des INPUT.

C'est le premier programme que j'ai fait. J'ai commencé à l'écrire le jour même ou j'ai reçu mon CBM il y a 3 mois. Je n'avais jamais eu accès à un ordinateur auparavant et j'ai appris le Basic en faisant ce programme. Donc pas d'ordinogramme et même pas de projet d'ensemble au départ puisque je ne savais pas ce que l'on pouvait faire avec un micro-ordinateur. Ceci vous explique le numérotage un peu serré puisqu'il m'a fallu serrer, serrer, renuméroter les lignes, au fur et à mesure que le programme se complétait (malheureusement Edex n'existe pas encore pour la série 4000).

Des amis qui ont des micro-ordinateurs depuis longtemps m'ont dit qu'un "truc" que j'ai trouvé en écrivant ce programme pouvait intéresser les lecteurs: l'utilisation d'un fichier dans le programme.

En effet, l'utilisation d'un fichier sur magnétophone m'a semblé très mal pratique :

- nécessité d'avoir 2 magnétos pour les mises à jour.

- nécessité de recopier à chaque ajout tout le fichier d'un magnéto à l'autre avec les risques d'erreur de copie que cela comporte (pas de Vérify sur les fichiers)

- nécessité de relire tout le fichier sur le magnéto à chaque nouvelle analyse du fichier.

Puisque les micro-ordinateurs actuels ont de la mémoire centrale en quantité, on peut mettre le fichier (s'il est petit ou moyen) dans le programme. Si le programme lui-même tourne avec 8K, il reste 8K ou 24K (CBM 4016 ou 4032) pour mettre le fichier dans des DATA. En créant des lignes de programme-fichier par des INPUT, on peut faire des ajouts, des modifications, des suppressions et recopier le tout sur le magnéto une fois terminé ; et ceci en bénéficiant de la fonction VERIFY (dans mon cas de l'indice INSEE du coût de la vie, on ne fait que des ajouts au fur et à mesure de la publication des indices)

Les lignes intéressantes pour la mise à jour du fichier sont :

. 8260 NL = N + 601
ou NL = DM + 277

C'est la détermination du numéro de ligne (NL) à générer selon que l'on désire ajouter des indices annuels (N) ou mensuels (DM).

. 8550 INPUT (les indices à ajouter)

. 8810 et 8840 Création d'une ligne

de programme de DATA avec pour numéro de ligne NL et pour contenu celui de la réponse à l'INPUT de la ligne 8550.

C'est donc très court (4 lignes de programme). Dans mon programme c'est très long à cause de toutes les explications affichées à l'écran, des vérifications des données entrées dans l'INPUT, de la possibilité d'entrer 1 ligne et demie d'indices à la fois.

Au lieu de DATA on pourrait aussi créer des lignes de programmes contenant une variable multiple. Exemple : 310 I(82) =..., I(83) =..., I(84) =..., mais cela occupe plus de place mémoire centrale et est plus long à programmer.

En fait, dans ce programme, les DATA sont retranscrits dans des tableaux qui me donnent un fichier à accès direct en mémoire centrale et un temps d'accès très faible, inférieur à celui des disquettes ; mais je n'ai pu faire ceci que parce que mon fichier est petit. J'ai d'ailleurs dans ce programme gaspillé la mémoire centrale au profit de la vitesse d'exécution (et de la limitation du temps de lecture de la cassette).

Le programme occupe 10600 octets avant de tourner et 12800 après avoir tourné. Si l'on supprimait les nombreuses explications apparaîssant à l'écran, il serait beaucoup plus court.

Peut-être le "truc" du fichier vous est-il déjà connu ; mais je ne l'ai trouvé cité dans aucun des bouquins et revues assez nombreux que j'ai consulté. Bien sûr, la création de lignes de programme est classique (bien que peu citée), mais je ne l'ai pas vue appliquée aux fichiers.

Dr. Alain TASSY

```
2 REM  ###   INFLATION   ###
3 REM ECRIT EN 12/1981 PAR ALAIN TASSY
6 REM ADRESSE:LA RESERVE RUE CONDORCET
7 REM LE MOURILLON 83000 TOULON
8 REM TEL. 46-64-72
11 REM PROGRAMME PREVU POUR POUVOIR AJOUTER LES INDICES JUSQU'EN
15 :                                                    1999 INCLUS
16 REM===SS-PRG. PLACE DU . + AUTRE
18 GOTO 45
19 X=INT(P*FI/DE*100+.5)/100
20 XA=ABS(X)
22 XN=LEN(STR$(INT(XA))):IF XA<1 AND XA>0 THEN XN=1
24 RETURN
45 REM=====PRG. PRINCIPAL
90 PRINT"▯      ####       ▮INFLATION▮      ####":PRINT
100 PRINT"   CE PROGRAMMME INDIQUE L'INFLATION EN FRANCE DEPUIS
110 PRINT"   A PARTIR DE 1970 ON UTILISE L'INDICE"    1901":PRINT
115 PRINT"INSEE DES PRIX DE DETAIL (FRANCE ENTIE-
120 PRINT"RE).":PRINT
125 PRINT"   AVANT 1970 ON UTILISE L'INDICE MOYEN
130 PRINT"(GROS ET DETAIL) PUBLIE PAR LE MINISTE-
135 PRINT"RE DE L'ECONOMIE. CET INDICE A ETE RE-
140 PRINT"CALCULE EN VALEUR INDICE INSEE":PRINT
150 PRINT"   TOUT EST CONVERTI EN NOUVEAUX FRANCS":PRINT
160 PRINT"   DANS LES TABLEAUX L'ANNEE EST INDI-
170 PRINT"QUEE PAR SES 2 DERNIERS CHIFFRES ET LE
180 PRINT"MOIS PAR SON NUMERO. EXEMPLE: 70 = 1970
185 PRINT"ET 5/80 = MAI 1980
190 PRINT"▮   CE QUE JE NE CONNAIS PAS EST INDIQUE PAR X
200 PRINT"▮▮POUR CONTINUER TAPEZ RETURN ET ATTENDEZ";
208 IF N<>0 GOTO 780:REM C'EST LE GET
```

```
220 DIM I$(101),B(16),IN$(35)
270 REM=====INDICES MENSUELS. NE PAS INTRODUIRE D'INDICE ANTERIEUR. LE NUMERO
271 REM DE LIGNE-276 CORRESPOND AU NU-
272 REM MERO DE MOIS A PARTIR DE 1/78
277 DATA 190.3,191.7,193.4,195.5,197.4,198.9:REM C'EST JANVIER 78 ETC
283 DATA 201.3,202.5,203.8,205.7,206.8,207.8
289 DATA 209.7,211.1,213,215.1,217.4,219.2:REM C'EST 1/79 ETC
295 DATA 222.1,224.4,226.3,229,230.5,232.3
301 DATA 236.8,239.3,242.1,244.9,247.1,248.7:REM C'EST 1/80 ETC
307 DATA 252.4,254.9,257.1,259.9,261.7,264
313 DATA 267.1,269.6,272.3,276.0,278.5,281.2:REM C'EST 1/81 ETC
319 DATA 286.1,289.6,292.9,296.5,299.2,301
544 DATA 0:REM CE 0 EST UN REPERE CAPITAL. LAISSER LIBRE JUSQU'EN 544 POUR IM
545 REM FUTURS
590 REM======INDICES ANNUELS. LES 2 DERNIERS CHIFFRES DU NO DE LIGNE INDIQUENT
592 REM L'ANNEE
600 DATA .2899,.2792,.2792,.2792,.2792,.29,.2792,.3007,.29,.29,.3007
611 DATA .3329,.3329,.3329,.3329,.4296,.537,.7088,.9129,1.02,1.44
621 DATA 1.085,1.042,1.246,1.439,1.59,2.051,1.933,1.944,1.976,1.858
631 DATA 1.675,1.514,1.45,1.375,1.278,1.428,1.89,2.148,2.277,2.835
641 DATA 3.405,4.038,4.844,5.681,8.238,13.35,20.12,33.34,37.45,40.85
651 DATA 49.96,53.95,52.22,51.84,52.05,54.26,56.66,64.14,67.63,69.72
661 DATA 71.58,74.27,77.46,79.28,80.98,83.2,84.2,86.86,93.41,100
671 DATA 105.5,112,120.2,136.7,152.8,167.5,183.2,199.8,221.3,251.3
700 DATA 0:REM CE 0 EST UN REPERE CAPITAL. LAISSER LIBRE JUSQU'EN 700 POUR I
701 REM FUTURS
705 REM======CALCULS
730 FOR DM=1 TO 270:READ V$
732 IF V$="0"THEN DM=DM-1:GOTO 740
734 NEXT
740 DIM M$(DM),IM$(DM):RESTORE
742 FOR Z=1 TO DM
744 A=INT((Z-1)/12):M=Z-(A*12):A=78+A
746 M$(Z)=RIGHT$(STR$(M),2)+"/"+RIGHT$(STR$(A),2)
748 READ IM$(Z):NEXT
760 READ A:REM PASSE LE 0
762 FOR Z=0 TO 102:READ I$(Z)
764 IF I$(Z)="0"THEN I$(Z)="":N=Z-1:GOTO 780
766 NEXT
780 GET A$: IF A$<>CHR$(13) GOTO 780
800 PRINT"❑     ####          ■MENU■          ####":PRINT:PRINT
```

```
805 PRINT" LES DERNIERS INDICES QUE JE CONNAIS
807 PRINT"SONT CEUX DE"1900+N"ET "M$(DM):PRINT
810 PRINT"▒1▒ INDICES ANNUELS DEPUIS 1901 (OU 88       DERNIERES ANNEES)":PRINT
820 PRINT"▒2▒ INDICES MENSUELS DEPUIS 1/78 (OU 63      DERNIERS MOIS)":PRINT
830 PRINT"▒3▒ INFLATION ANNUELLE DEPUIS 1901 (OU 87    DERNIERES ANNEES)
840 PRINT"▒▒4▒ INFLATION MENSUELLE DEPUIS ";M$(1)
850 PRINT"▒▒5▒ VALEUR D'UNE CHOSE EN L'ANNEE ▒Y▒ EN"
855 PRINT"  FONCTION DE SON PRIX EN L'ANNEE ▒X▒":PRINT
895 PRINT"▒6▒ ENTREE D'UN NOUVEL INDICE
910 PRINT"▒▒▒
```

```
3050 PRINT"▒  JE PEUX L'AFFICHER SIMULTANEMENT POUR 36 MOIS CONSECUTIFS
3060 PRINT"▒  INDIQUEZ-MOI LE DERNIER MOIS QUE VOUS DESIREZ
3070 INPUT "▒▒NUMERO DU MOIS (SANS L'ANNEE)▒";M
3072 INPUT "▒▒DEUX DERNIERS CHIFFRES DE L'ANNEE▒";A
3076 M=(A-78)*12+M:IF M<1 OR M>DM THEN PRINT"▒VOUS SORTEZ DES LIMITES":GOTO3070
3082 PRINT"▒  LE MOIS EST INDIQUE PAR NO DU MOIS ET DERNIER CHIFFRE DE L'ANNEE
3084 PRINT"  LA 1ER COLONNE INDIQUE LE MOIS; LA 2EM";
3086 PRINT"L'INFLATION MENSUELLE; LA 3EM L'INFLA-
3088 PRINT"TION PAR RAPPORT AU MEME MOIS DE L'ANNEE";
3090 PRINT"PRECEDENTE
3109 PRINT"▒▒▒
```

```
4070 PRINT"CONVERTISSEZ-LE EN NOUVEAUX FRANCS.":PRINT
4080 PRINT" NE TAPEZ QUE LES CHIFFRES ET PAS LE MOT";
4090 PRINT"FRANC. ▒S'IL Y A UNE VIRGULE REMPLACEZ-LA";
4100 PRINT"▒PAR UN POINT▒":PRINT
4120 PRINT" JE PEUX CALCULER SEULEMENT POUR LES AN-";
4130 PRINT"NEES 1901 A";STR$(N+1900);". MAIS JE DONNERAI EN
4133 PRINT"PLUS LA VALEUR POUR LE DERNIER MOIS QUE
4136 PRINT"JE CONNAIS ("M$(DM)")."":PRINT
4140 PRINT" APRES CHAQUE VALEUR TAPEZ TOUCHE RETURN"
4150 INPUT "▒CETTE CHOSE VALAIT▒";P:PRINT
4155 PRINT"▒EN L'ANNEE (2 DERNIERS CHIFFRES DE L'
4157 PRINT"▒ANNEE). OU EN "M$(DM)" (TAPEZ M)▒";:INPUT "▒▒▒▒*▒▒▒▒";A$
4159 PRINT"▒                                                    "
4161 IF A$="*"THEN PRINT"▒▒▒▒":GOTO4157
4175 IF A$="M"THEN DE=VAL(IM$(DM)):B$=M$(DM):GOTO 4230
4185 B$=RIGHT$(A$,2):IF VAL(B$)<=N AND I$(VAL(B$))<>"X"GOTO 4190
4187 PRINT"▒L'INDICE DE 19"B$" ME MANQUE▒▒":GOTO 4157
4190 DE=VAL(I$(VAL(B$))):B$=STR$(VAL(B$)+1900)
4230 INPUT "▒▒VOUS VOULEZ CONNAITRE SA VALEUR EN       FRANCS DE L'ANNEE▒";A
4240 A=VAL(RIGHT$(STR$(A),2))
4245 IF I$(A)="X"OR I$(A)=""THEN V$=": CET INDICE ME MANQUE":GOTO 4300
4250 FI=VAL(I$(A)):GOSUB 19:REM CALCULS
4260 V$=STR$(X)+" FRANCS"
4300 PRINT"▒UN PRIX DE ▒"P"FRANCS▒":PRINT
4310 PRINT"▒EN "B$"▒ CORRESPOND EN FRANCS DE :":PRINT
4350 PRINT"▒"A+1900"▒A ▒"V$"▒":PRINT
4370 B(0)=1:B(1)=14:B(2)=20:B(3)=39:B(4)=48:B(5)=60:B(6)=70:B(7)=N-9:B(8)=N-8
4380 B(9)=N-7:B(10)=N-6:B(11)=N-5:B(12)=N-4:B(13)=N-3:B(14)=N-2:B(15)=N-1
4385 B(16)=N
4390 FOR Z=0 TO 16
4395 E=0:IF Z>8 THEN E=21:IF Z=9 THEN PRINT"▒▒▒▒▒▒"
4410 IF I$(B(Z))="X"THEN V$="X":XN=0:GOTO 4470
4420 FI=VAL(I$(B(Z))):GOSUB19:V$=STR$(X)
4470 PRINTTAB(E)B(Z)+1900;"A";TAB(15+E-XN)V$:PRINT
4510 NEXT
4570 FI=VAL(IM$(DM)):GOSUB 19
4582 PRINT"▒"TAB(E)M$(DM);" A";TAB(15+E-XN)STR$(X)
4650 PRINT"▒▒▒▒▒▒
```

```
5050 DEBUT=1:IF DM>63 THEN DE=DM-62
5060 FOR Z=DE TO DM:E=INT((Z-DE)/21)*13
5070 IF (Z-DE)/21=INT((Z-DE)/21) THEN PRINT"■■"
5080 PRINTTAB(E)"|";M$(Z);TAB(E+8)IM$(Z):NEXT
5150 GET A$:IF A$="2" GOTO 800
5160 IF A$="3" GOTO 90
5170 GOTO 5150
5500 REM======TABLEAU INDICES ANNUELS
5540 PRINT"■|AN INDICE|AN INDICE|AN INDICE|AN INDIC"
5550 PRINT"■■■
```

```
8375 PRINT"■■■■■TAPER (RETURN)
8380 GET A$:IF A$<>CHR$(13)GOTO 8380
8400 PRINT"■POUR ABANDONNER TAPEZ 2 ET TOUCHE RETURN
8410 PRINT" LE DERNIER INDICE ■"B$"■ QUE JE CON-":PRINT
8420 PRINT"NAIS EST CELUI DE "V$" ="E:PRINT
8450 PRINT" TAPEZ LES INDICES SOUS LA FORME :":PRINT
8460 PRINT"- EXEMPLE POUR 1 INDICE: 238.5":PRINT
8470 PRINT"- EXEMPLE POUR 3 I.: 892.5/956.6/1024
8550 POKE 158,0:INPUT "■■■■■■■■■■■■■■■INDICE(S)";A$
8555 IF A$="2"GOTO 800
8557 PRINT"■■■■■■■■■■■■■■■
```

Calcul du déterminant d'une matrice carrée

La méthode employée consiste à transformer une matrice M de départ autant de fois qu'il le faut, jusqu'à l'obtention d'une matrice triangulaire:

$$\begin{matrix} e_{11} & & e_{n1} \\ \cdot & \cdot & \\ \cdot & \cdot & \\ \cdot & \cdot & \\ e_{1n} & & e_{nn} \end{matrix} \quad \longrightarrow \quad \begin{matrix} e'_{11} & & & & e'_{n1} \\ 0 & e'_{22} & & & \cdot \\ 0 & 0 & \cdot & & \cdot \\ \cdot & \cdot & & \cdot & \cdot \\ 0 & 0 & \ldots & 0 & e'_{nn} \end{matrix}$$

Le déterminant det(M) de M est alors égal à:

$$\det(M) = e'_{11} * e'_{22} * \ldots * e'_{nn} * k$$

ou:

-les e'_{ii} sont les éléments diagonaux de la matrice,

-f est un coefficient dont la valeur est modifiée à chaque transformation de la matrice.

* transformations invariantes:

Ce sont les transformations qui ne modifient pas la valeur de f. En particulier: l'addition ou la soustraction de 2 colonnes ou de 2 lignes.

* transformations non invariantes:

- échange de 2 lignes ou de 2 colonnes: f change de signe.

- multiplication d'une ligne ou d'une colonne par un scalaire p: f doit être divisé par p.

exemple:

$$DE = \det \begin{matrix} 1 & 3 \\ 2 & 4 \end{matrix} = \det \begin{matrix} 2 & 3 \\ 4 & 4 \end{matrix} * 1/2$$

$$= \det \begin{matrix} -1 & 3 \\ 0 & 4 \end{matrix} * 1/2 = -1 * 4 * 1/2$$

$$= -2$$

Dans cet exemple 2 transformations élémentaires ont permis d'obtenir une matrice triangulaire. La première transformation consiste à multiplier la première colonne par 2: elle n'est pas invariante. Par contre la seconde l'est, qui consiste à soustraire la 2ème colonne à la 1ère.

Dans le programme ci-après:

- le tableau M(nc,nl) contient les valeurs de la matrice,

- M sert de variable de transfert lors des échanges entre colonnes,

- I, J, L sont les variables de boucle permettant de désigner les éléments de la matrice concernés par l'opération.

```
0 rem * calcul determinant d'une matrice carree *
1 print"■": f=1: de=1
10 input"Nbre de colonnes ";nc: nc=nc-1: nl=nc: dim m(nc,nl)
20 for i=0 to nc: for j=0 to nl
30 print"Valeur:  col"i+1" ligne"j+1;: input m(i,j)
40 next j,i
45 for i=0 to nc-1
50 for j=nl to i+1 step-1
55 a=m(i,j): b=m(i,j-1)
60 if a=0 then next j,i: goto 120
65 if b then 80
70 for l=0 to nc: m=m(l,j): m(l,j)=m(l,j-1): m(l,j-1)=m: next: f=-f
75 nextj: for j=0 to nl: m=m(l,j): m(l,j)=m(l+1,j): m(l+1,j)=m: next
77 f=-f: goto 50
80 p=-b/a
100 for l=i to nc: m(l,j)=m(l,j)*p: next
110 for l=i to nc: m(l,j)=m(l,j)+m(l,j-1): next: f=f/p: next j,i
120 for i=0 to nl: de=de*m(i,i): next: de=de*f
130 print"Det(m)="de
140 print"■■Autre determinant ?";:poke167,0:getr$
150 if r$="" then get r$: goto 150
160 if r$="o" then run
170 end
ready.
```

A VOS CLAVIERS !

PROCEP recherche programmeurs sachant l'anglais pour franciser à domicile des logiciels sur COMMODORE 64.

Les candidats sélectionnés devront effectuer le travail dans un temps limité et être déjà très familiers avec la programmation BASIC et Assembleur sur COMMODORE. En contre-partie, ils pourront travailler avec les derniers matériels de COMMODORE et vivre la Révolution Micro-informatique en temps réel.

Adresser CV détaillé à B. CHALLET, PROCEP, 19-21 rue Mathurin-Régnier 75015 PARIS.

Routines-Système du 64

N.B. On indique les points de début
--- des routines. Ce ne sont pas
forcément les points d'entrée.

HEXA	DEC	
A000	40960	vrs (vecteurs) ROM
A00C	40972	vrs action des mots-clés
A052	41042	vrs fonctions
A080	41088	vrs opérations
A09E	41118	mots clés
A19E	41374	messages d'erreur
A328	41178	vrs messages d'erreur
A365	41829	messages divers
A38A	41866	balaie la pile pour FOR / GOSUB
A3B8	41912	translation en mémoire
A3FB	41979	vérifie profondeur pile
A408	41992	vérifie espace mémoire
A435	42037	"OUT OF MEMORY"
A437	42039	routine d'erreur
A469	42089	entrée BRK
A474	42100	"READY"
A480	42112	préparation pour basic
A49C	42140	traite nouvelle ligne
A533	42291	rétablit le chaînage
A560	42336	reçoit ligne d'entrée
A579	42361	remplace les mots clés par leur code
A613	42515	trouve ligne BASIC
A642	42562	effectue NEW
A65E	42590	effectue CLR
A68E	42638	copie pointeur texte
A69C	42652	effectue LIST
A742	42818	effectue FOR
A7ED	42989	effectue instruction
A81D	43037	effectue RESTORE
A82C	43052	arrêt
A82F	43055	effectue STOP
A831	43057	effectue END
A857	43095	effectue CONT

HEXA	DEC	
A871	43121	effectue RUN
A883	43139	effectue GOSUB
A8A0	43168	effectue GOTO
A8D2	43218	effectue RETURN
A8F8	43256	effectue DATA
A906	43270	recherche prochaîne instruction
A928	43304	effectue IF
A93B	43323	effectue REM
A94B	43339	effectue ON
A96B	43371	prend un nombre entier
A9A5	43425	effectue LET
AA80	43648	effectue PRINT
AA86	43654	effectue CMD
AAA0	43680	effectue PRINT
AB1E	43806	imprime message pointé par (Y,A)
AB3B	43835	impr caractère de format
AB4D	43853	routine mauvaise entrée
AB7B	43899	effectue GET
ABA5	43941	effectue INPUT
ABBF	43967	effectue INPUT
ABF9	44025	interroge pour INPUT
AC06	44038	effectue READ
ACFC	44284	mesg d'erreur en entrée
AD1E	44318	effectue NEXT
AD78	44408	vérifie accord de type
AD9E	44446	effectue expression
AEA8	44712	constante "PI"
AEF1	44785	effectue les parenthèses
AEF7	44791	vérifie ")"
AEFF	44799	vérifie ","
AF08	44808	erreur de syntaxe
AF14	44820	vérifie domaine
AF28	44840	cherche variable
AFA7	44967	référence à FN...
AFE6	45030	effectue OR
AFE9	45033	effectue AND
B016	45078	comparaison
B081	45185	effectue DIM

HEXA	DEC	
B08B	45195	localise variable
B113	45331	vérif car. alphabétique
B11D	45341	crée une variable
B194	45460	sp pointeur vers tableau
B1A5	45477	valeur 32768
B1B2	45490	convers. flottant/entier
B1D1	45521	prépare tableau
B245	45637	"BAD SUBSCRIPT"
B248	45640	"ILLEGAL QUANTITY"
B34C	45900	calcule taille tableau
B37D	45949	effectue FRE
B391	45969	conv. entier-flottant
B39E	45982	effectue POS
B3A6	45990	vérifie mode direct
B3B3	46003	effectue DEF
B3E1	46049	vérifie syntaxe de FN
B465	46181	effectue STR$
B475	46197	calcule vect de chaîne
B487	46215	prépare chaîne
B4F4	46324	fait de la place pour une chaîne
B526	46374	"garbage collection"
B5BD	46525	teste si à conserver
B606	46598	rassemble chaîne
B63D	46653	concaténation
B67A	46714	construit chaîne en mémoire
B6A3	46755	rejette chaîne
B6DB	46811	nettoie la pile des descripteurs
B6EC	46828	effectue CHR$
B700	46848	effectue LEFT$
B72C	46892	effectue RIGHT$
B737	46903	effectue MID$
B761	46945	dépile les paramètres de chaîne
B77C	46972	effectue LEN
B782	46978	sort du mode chaîne
B78B	46987	effectue ASC
B79B	47003	obtient octet paramètre
B7AD	47021	effectue VAL
B7EB	47083	obtient les paramètres pour POKE/WAIT
B7F7	47095	conv. flottant-entier
B80D	47117	effectue PEEK
B824	47140	effectue POKE
B82D	47149	effectue WAIT
B849	47177	ajoute 0.5
B850	47184	opérande de soustraction
B853	47187	effectue soustraction
B86A	47210	effectue addition
B947	47431	complémente ACC1
B97E	47486	"OVERFLOW"
B983	47491	multiplie par octet nul
B9EA	47594	effectue LOG
BA2B	47659	effectue multiplication
BA59	47705	multiplication d'un bit
BA8C	47756	mémoire vers ACC2
BAB7	47799	ajuste ACC1/ACC2
BAD4	47828	dépassement/sous-dép.

HEXA	DEC	
BAE2	47842	multiplie par 10
BB12	47890	effectue division
BBA2	48034	mémoire vers ACC1
BBC7	48071	ACC1 vers mémoire
BBFC	48124	ACC2 vers ACC1
BC0C	48140	ACC1 vers ACC2
BC1B	48155	arrondit ACC1
BC2B	48171	obtient le signe
BC39	48185	effectue SGN
BC58	48216	effectue ABS
BC5B	48219	compare ACC1 et mémoire
BC9B	48283	conv. flottant-entier
BCCC	48332	effectue INT
BCF3	48371	chaîne vers ACC
BD7E	48510	prend chiffre ASCII
BF16	48918	constantes décimales
BF3A	48954	constantes pour TI
BF71	49009	effectue SQR
BF7B	49019	effectue "puissance"
BFB4	49076	effectue "opposé"
BFED	49133	effectue EXP
E043	57411	évaluation de série (1)
E059	57433	évaluation de série (2)
E097	57495	effectue RND
E0F9	57593	points d'arrêt
E12A	57642	effectue SYS
E156	57686	effectue SAVE
E165	57701	effectue VERIFY
E168	57704	effectue LOAD
E1BE	57790	effectue OPEN
E1C7	57799	effectue CLOSE
E1D4	57812	paramètres pour LOAD/SAVE
E206	57862	vérifie param par défaut
E20E	57870	teste virgule
E219	57881	paramètres pour OPEN/CLOSE
E264	57957	effectue COS
E26B	57963	effectue SIN
E2B4	58036	effectue TAN
E30E	58126	effectue ATN
E37B	58235	reset tiède
E394	58260	initialisation
E3A2	58274	routine CHRGET (sera copiée en page 0)
E3BF	58303	initialise BASIC
E447	58439	vecteurs /copiés en $300
E453	58451	initialise les vecteurs
E45F	58463	message à la mise sous tension
E500	58624	prépare adresses E/S
E505	58629	établit limites écran
E50A	58634	met/obtient ligne/col
E518	58648	initialise E/S
E544	58692	vide écran
E566	58726	curseur à l'origine
E56C	58732	établit pointeurs écran
E5A0	58784	prépare val par défaut des E/S
E5B4	58804	entrée depuis le clavier

HEXA	DEC	
E632	58930	entrée depuis l'écran
E684	59012	teste guillemet
E691	59025	prépare impression écran
E6B6	59162	avance curseur
E6ED	59117	recule curseur
E701	59137	recule dans la ligne précédente
E716	59158	sortie écran
E87C	59516	va à la ligne
E891	59597	retour chariot
E8A1	59553	vérifie diminution de ligne
E8B3	59571	vérifie augmentation de ligne
E8CB	59595	établit code couleur
E8DA	59610	table des codes couleur
E8EA	59726	déroulement écran
E965	59749	fait place sur l'écran
E9C8	59851	déplace ligne sur écran
E9E0	59872	transfert synchro couleur
E9F0	59888	établit début de ligne
E9FF	59903	vide ligne écran
EA13	59923	imprime sur écran
EA24	59940	synchro couleur caract.
EA31	59953	interruption IRQ
EA87	60039	teste clavier
EB79	60281	vecteurs clavier
EB81	60289	carte clavier
EC44	60484	controle graphique/texte
EC4F	60495	met en mode graphique
EC78	60536	matrice des touches avec CTRL
ECB9	60601	options par défaut 6567
ECF0	60656	adr. basse ligne écran
ED09	60681	envoie "parle"
EDOC	60684	envoie "écoute"
ED25	60709	envoie carac de commande
ED40	60736	envoi sur bus série
EDB2	60850	time-out sur bus série
EDB9	60857	envoie adresse secon-
		daire écouteur
EDBE	60862	RAZ ATN
EDC7	60871	envoie adresse secon-
EDCC	60876	attente horloge
		daire parleur
EDDD	60893	envoi série différé
EDEF	60911	env. cessation émission
EDFE	60926	envoie cessation écoute
EE13	60947	réception sur bus série
EE85	61061	met à 1 ligne horloge
EE8E	61070	met à 0 ligne horloge
EE97	61079	sortie série à 1
EEA0	61088	sortie série à 0
EEA9	61097	acquiert donnée+horloge
EEB3	61107	délai 1 ms
EEBB	61115	envoi sur RS232 (NMI)
EF06	61190	envoie nouvel oct. RS232
EF2E	61230	erreur: modem non prêt
EF31	61233	err.: pas prêt à émettre

HEXA	DEC	
EF3B	61243	désactive temporisateur
EF4A	61258	compte de bits
EF59	61273	reçoit sur RS232 (NMI)
EF7E	61310	prépare réception
EFC5	61381	err parité en réception
EFCA	61386	err surcharge réception
EFCD	61389	err rupture en réception
EFD0	61392	err cadrage en réception
EFE1	61409	fichier vers RS232
F017	61463	envoi vers tampon RS232
F04D	61517	INPUT depuis tamp RS232
F086	61574	GET depuis tampon RS232
F0A4	61604	vérif bus série au repos
F0BD	61629	messages
F12B	61739	imprime si mode direct
F13E	61758	lecture ...
F14E	61774	sur RS232
F157	61783	entrée
F199	61849	GET cassette/série/RS232
F16A	61898	sortie ...
F1DD	61917	sur cassette
F20E	61966	établit périph d'entrée
F250	62032	établit périph de sortie
F291	62097	CLOSE
F30F	62223	trouve fichier
F31F	62239	prépare fichier
F32F	62255	supprime ts les fichiers
F333	62259	rétablit périph standard
F34A	62282	ouvre fichier
F3D5	62421	envoie adr. secondaire
F409	62473	ouvre RS232
F49E	62622	charge programme
F5AF	62895	"SEARCHING"
F5C1	62913	imprime nom de fichier
F5D2	62930	"LOADING/VERIFYING"
F50D	62941	sauve programme
F68F	63119	"SAVING"
F69B	63131	incrémente horloge
F6BC	63164	note la touche lue sur PIA
F6DD	63197	prend l'heure
F6E4	63204	met à l'heure
F6ED	63213	obéit à la touche STOP
F6FB	63227	mess d'err sur fichiers
F72D	63277	trouve en-tête bande
F76A	63338	écrit en-tête bande
F7D0	63440	obtient adresse tampon
F7D7	63447	établit pointeurs début/fin tampon
F7EA	63466	trouve en-tête spécifique
F80D	63501	incrém. pointeur bande
F817	63511	"PRESS PLAY ..."
F82E	63534	teste état cassette
F838	63544	"PRESS RECORD ..."
F841	63553	commence lecture bande
F864	63588	commence écriture bande
F875	63605	code commun lect./écrit.
F8D0	63696	vérifie arrêt bande
F8E2	63714	établit le timing

RUN INFORMATIQUE
62, rue Gérard 75013 PARIS (1) 581-51-44

LE SPECIALISTE DES LOGICIELS DE JEUX, DES LOGICIELS SERIEUX ET DES EXTENSIONS
pour le VIC 20 et le CBM 64

de nouveaux jeux (K7)

VIC 20
- TIME DESTROYER (3, 8, 16 K) ... réf. 2022 145 F
- ATOM-SMASHER ... 2021 145 F
- GALACTIC ABDUCTOR (16 K) ... 9002 145 F
- MATRIX (8-16 K) ... 10006 135 F
- LAZER ZONE ... 10004 135 F

CBM 64
- LAZER ZONE ... 10054 155 F
- STIX ... 6021 155 F
- MATRIX ... 10005 155 F
- GRAND MASTER *super jeu d'échecs* ... 6020 290 F

du sérieux

CBM 64
- VIC TREE ... 6090 975 F
- GRAPHIX (K7) 6065 : 195 F — disquette 6565 235 F
- ZOOM ... (K7) 6050 : 195 F — disquette 6550 235 F
- ARROW ... 6060 670 F

BUSICALC *un vrai "calc" professionnel!*
- VIC 20 ... 6001 600 F
- COMMODORE 64 *disquette* ... 6002 630 F
- CBM 4032 *disquette* ... 6003 630 F
- CBM 8032 *disquette* ... 6004 630 F

des extensions

VIC 20 / CBM 64
- 32 K ... 3003 849 F
- 16 K ... 8002 549 F
- CARTE MERE 3 connecteurs ... 8100 349 F

... et bientôt des jeux français

BON DE COMMANDE
à découper et à renvoyer à : RUN INFORMATIQUE
62, rue Gérard 75013 Paris

M _____
Adresse _____
_____
Matériel _____

Je passe commande de : *Total*
LOGICIEL JEUX N° ____ N° ____ N° ____ ____
LOGICIEL GESTION N° ____ N° ____ N° ____ ____
BUSICALC N° ____ ____
EXTENSION N° ____ ____
Ci-joint mon règlement par chèque bancaire ☐
ou CCP ☐

☐ Je désire recevoir votre documentation
SIGNATURE:

Tous les prix comprennent la T.V.A. et les frais de port postaux
(France Métropolitaine)

Oct 83 pyc publicité

HEXA	DEC	
F92C	63788	lit bits (IRQ)
FA60	64096	range caractères
FB8E	64398	RAZ pointeur
FB97	64407	prépare nouveau caract.
FBA6	64422	commute bande
FBC8	64456	écrit données
FBCD	64461	écrit sur bande (IRQ)
FC57	64599	écrit en-tête (IRQ)
FC93	64659	restaure vecteurs
FC38	64696	établit vecteur
FCCA	64714	arrête moteur
FCD1	64721	teste pointeur lecture/écriture
FCDB	64731	incrémente pointeur
FCE2	64738	entrée mise sous tension
FD02	64770	teste si ROM en A000
FD10	64784	masque ROM 8000
FD15	64789	rétablit noyau
FD1A	64794	déplace noyau
FD30	64816	vecteurs
FD50	64848	init. constantes système
FD9B	64923	vecteurs IRQ
FDA3	64931	initialise registres E/S
FDF9	65017	sauve nom des données
FE00	65024	sauve détails fichiers
FE07	65031	prend l'état
FE1C	65052	marque ST
FE21	65057	marque time-out
FE25	65061	cherche/établit haut de mémoire
FE27	65063	lit haut de mémoire
FE2D	65069	établit haut de mémoire
FE34	65076	cherche/établit bas de mémoire
FE43	65091	entrée interruption NMI
FE66	65126	départ à chaud RESET/STOP
FEB6	65206	restaure IRQ et sort
FEBC	65212	sortie IRQ
FEC2	65218	table de temps RS232
FED6	65238	entrée RS232 (NMI)
FF07	65287	sortie RS232 (NMI)
FF43	65347	fausse IRQ
FF48	65352	entrée IRQ principale
FF81	65409	table de vecteurs dont:
FFC6	65478	établit canal d'entrée
FFC9	65481	établit canal de sortie
FFCC	65484	restaure canaux d'E/S standard
FFCF	65487	entrée (INPUT)
FFD2	65490	sortie (PRINT)
FFE1	65505	teste touche STOP
FFE4	65508	saisie (GET)
FFFA	65530	vecteurs hardware.

A la Recherche des Routines Perdues

L'article est consacré à l'analyse d'un court programme en langage machine, que l'auteur a été amené à écrire pour rechercher, sur CBM 4032, les adresses d'un certain nombre de routines internes.

En effet la documentation donne de nombreuses adresses sur anciennes ROMs, sur la version 3000 et 8000, mais est souvent imprécise, voire muette, sur la version 4000.

Introduction

Lorsque l'on change de version de CBM (passage d'un 3032 à un 4032 par changement de ROMS, par exemple) les routines langage machine ne sont plus aux mêmes adresses. Comment trouver les nouvelles adresses connaissant les anciennes ?

Par exemple, avec un 3032 pour effacer l'écran, on peut faire JSR $E229 ou SYS 57897. Quid avec un 4032 ? (Non ! ce n'est pas indiqué ni dans "la pratique du PET/CBM Vol. II" ni dans "les clefs pour le PET/CBM".)

Le petit programme langage machine ci-après, baptisé "recherche LM", aide à résoudre ce problème. Il suppose connu le début de la routine recherchée : il suffit de lire, dans la version CBM que l'on connaît, le contenu des mémoires par des PEEK, ou en utilisant le moniteur L.M., ou dans le cas d'un 3032, de consulter "les programmes internes du PET/CBM" de B. MICHEL.

Dans notre exemple, à partir de l'adresse $E229 d'un 3032, on trouve : A0 83 A2 18 94 E0 ...

L'idée du programme est de balayer les mémoires jusqu'à trouver une suite d'octets identique à celle des N premiers octets recherchés. (7 octets au maximum).

Structure Générale du Programme

On utilise la zone mémoire $027A (634) à $02A0 (672) dans le tampon de la 1ère cassette. (pour changer... et parce que celui de la 2ème cassette contient déjà un autre programme !).

$027A contient le nombre N d'octets (7 maximum) de la chaîne à retrouver, elle-même placée à partir de $027B (cf. tableau 3).

Le programme proprement dit débute en $0282 (642).

Les octets $0285 et $0286 sont chargés initialement avec la partie basse et la partie haute de l'adresse de début de recherche : le programme examine si les N octets à partir de cette adresse sont identiques à ceux de la chaîne recherchée : sinon, l'adresse de début est incrémentée et la vérification est reprise jusqu'à trouver la coïncidence.

Détail du Programme

Le tableau 1 décrit, octet par octet, le déroulement du programme: la colonne "code" contient les codes POKE pour chargement en BASIC.

Ici l'adresse de départ de recherche est $B000. Les octets sont comparés deux à deux : on uti-

lise l'adressage indexé (Y). Le registre (Y), mis à zéro au début, est incrémenté si la comparaison réussit, jusqu'à atteindre le nombre d'octets indiqué dans $027A. Si toutes les comparaisons réussissent le programme s'arrête en $0292 (RTS=retour au BASIC) et l'adresse recherchée se trouve en $0285 et $0286, sinon, on va jusqu'à l'adresse $FFFF et le programme s'arrête en $029D. L'incrémentation de l'adresse s'effectue en deux temps: poids faible, puis, si nécessaire, poids fort.

Le tableau 2 présente le même programme sous forme assembleur plus classique.

Utilisation

Charger le programme en mémoire, soit par POKE des codes du tableau 1, à partir de 642, soit directement en hexa, par le moniteur langage machine.

Charger également en $027A (634) le nombre d'octets de la chaîne à rechercher (ici 6) puis les octets eux mêmes (ici : A0 B3 A2 18 94 E0, début de la routine d'effacement écran).

Mettre en $0285 la partie basse, et en $0286 la partie haute de l'adresse de début de recherche (ici $B000), sortir du moniteur par "X", et faire "SYS 642".

Le programme s'arrête après une fraction de seconde. (durée maxi, dans le plus mauvais des cas: 2.5 sec.).

On rappelle le moniteur par SYS 4, et on lit l'adresse cherchée (ici $E04B) en $0286 (partie haute) et $0285 (partie basse).

REMARQUE 1: Si on chargeait initialement l'adresse $0000, le programme s'arrêterait immanquablement à $02B7 (devinez pourquoi !) : on peut alors recommencer à partir de $027C.

REMARQUE 2: Attention ! une même chaîne, si elle est courte, peut se retrouver en plusieurs endroits. On peut aussi ne rien trouver du tout, (et dans ce cas, on lit $0000 en $0285 et $0286) essentiellement si la chaîne extraite d'une autre version CBM contient des JSR ou des JMP à des adresses différentes ! On doit alors choisir un tronçon de chaîne voisine, sans JSR ($20) ni JMP (4C).

REMARQUE 3: Toutes ces opérations s'effectuent facilement par le moniteur. Mais, si l'on préfère, il est aisé d'associer un programme BASIC conversationnel, pour charger le nombre d'octets par POKE 634,N, puis les octets eux-mêmes, par POKE 635,D1 : POKE 636,D2... puis de lire le résultat par:

PEEK(646)+256*PEEK(645) .

REMARQUE 4: On peut, bien sûr, faire tout en BASIC, en utilisant une boucle du type :

```
100  FOR I=A1 TO A2
110  IF PEEK(I)=D0 AND PEEK(I+1)=D1
     AND PEEK(I+2) AND PEEK(I+3)=D3
     THEN PRINT "ADRESSE=";I:END
120  NEXT
130  PRINT "Chaîne inexistante"
```

C'est simple, ça marche, c'est un peu long (5, voire 10 minutes), mais surtout, c'est beaucoup moins instructif !

Quelques Résultats sur CBM 4000-P

Effacement écran : SYS 57419 ($E04B)

Défilement de l'écran vers le haut ("scrolling up"): SYS 58217 ($E379)

Défilement vers le bas ("scrolling down"): SYS 58338 ($E3E2) (partiellement, à partir de la ligne où se trouve le curseur !)

STROUT : $BB1D (comme CBM 8000)- (sortie chaîne pointée par Y,A)

CRLF : $BAE1 (imprime retour chariot)

ADOUT : $D717 (comme CBM 8000)- (imprime $FB-FC en 4 chiffres hexa)

AOUT : $D722 (comme CBM 8000)- (imprime le contenu de (A) en 2 chiffres hexa).

AIN2 : $D7B5 : met dans (A) 2 chiffres hexa entrés au clavier et suivi de "RETURN". (Meilleur que $D754 qui suppose que les deux chiffres ont été précédés de deux espaces).

RDT : $FFCC (toutes versions) : attend que l'on frappe un caractère, puis "RETURN", et envoie son code ASCII en (A). Si plusieurs caractères sont frappés, il prend le premier

Tableau 1

J. PIERRAT

```
ETIQUETS.  MEMOIRES  MNEMOS.     CODES
========   ========  =======     =====

#DEBUT     0282 642  LDY IM       160
           0283 643  0            0
#LOOP      0284 644  LDA(Y)       185
           0285 645  $B000   (*)  0
           0286 646          (*)  176
           0287 647  CMP(Y)       217
           0288 648  $027B        123
           0289 649               2
           028A 650  BNE          208
           028B 651  AD.SUIV.     7
           028C 652  INY          200
           028D 653  CPY          204
           028E 654  $027A        122
           028F 655               2
           0290 656  BNE          208
           0291 657  LOOP         242
           0292 658  RTS(TROUVE!) 96
#AD.SUIV.  0293 659  INC          238
           0294 660  $0285        133
           0295 661               2
           0296 662  BNE          208
           0297 663  6            6
           0298 664  INC          238
           0299 665  $0286        134
           029A 666               2
           029B 667  BNE          208
           029C 668  1            1
           029D 669  RTS(PAS TROUVE!) 96
           029E 670  SEC          56
           029F 671  BCS          176
           02A0 672  DEBUT        225
```

BOUCLE DE COMPARAISON DE Y OCTETS 2 à 2

CHARGEMENT TERMINE DE 642 A 672

SOIT EN HEXA, DE 0282 A 02A0 :

```
0282  A0 00 B9 00 B0 D9 7B 02
028A  D0 07 C8 CC 7A 02 D0 F2
0292  60 EE 85 02 D0 06 EE 86
029A  02 D0 01 60 38 B0 E1 00
```

POUR EXECUTER, FAIRE SYS 642

OU UTILISER A=USR(X) APRES AVOIR FAIT:

POKE 1, 130 ET POKE 2, 2

NOMBRE D'OCTETS : 31

NOMBRE D'ETIQUETTES: 3

Tableau 2

```
0282 642   A0 00       LDYIM 0
0284 644   B9 ** **    LDAY  *
0287 647   D9 7B 02    CMPY  635
028A 650   D0 07       BNE   659
028C 652   C8          INY
028D 653   CC 7A 02    CPY   634
0290 656   D0 F2       BNE   644
0292 658   60          RTS
0293 659   EE 85 02    INC   645
0296 662   D0 06       BNE   670
0298 664   EE 86 02    INC   646
029B 667   D0 01       BNE   670
029D 669   60          RTS
029E 670   38          SEC
029F 671   B0 E1       BCS   642
```

Tableau 3

```
     PC   IRQ  SR AC XR YR SP
.:   0005 0382 30 00 5E 04 F8
.M 027A 02A0
.:   027A 06 A0 83 A2 18 94 E0 00
.:   0282 A0 00 B9 ** ** D9 7B 02
.:   028A D0 07 C8 CC 7A 02 D0 F2
.:   0292 60 EE 85 02 D0 06 EE 86
.:   029A 02 D0 01 60 38 B0 E1 00
.X
```

$027A : NOMBRE D'OCTETS DE LA CHAINE

$027B A 0281 : CHAINE D'OCTETS A RETROUVER (MAXI 7)

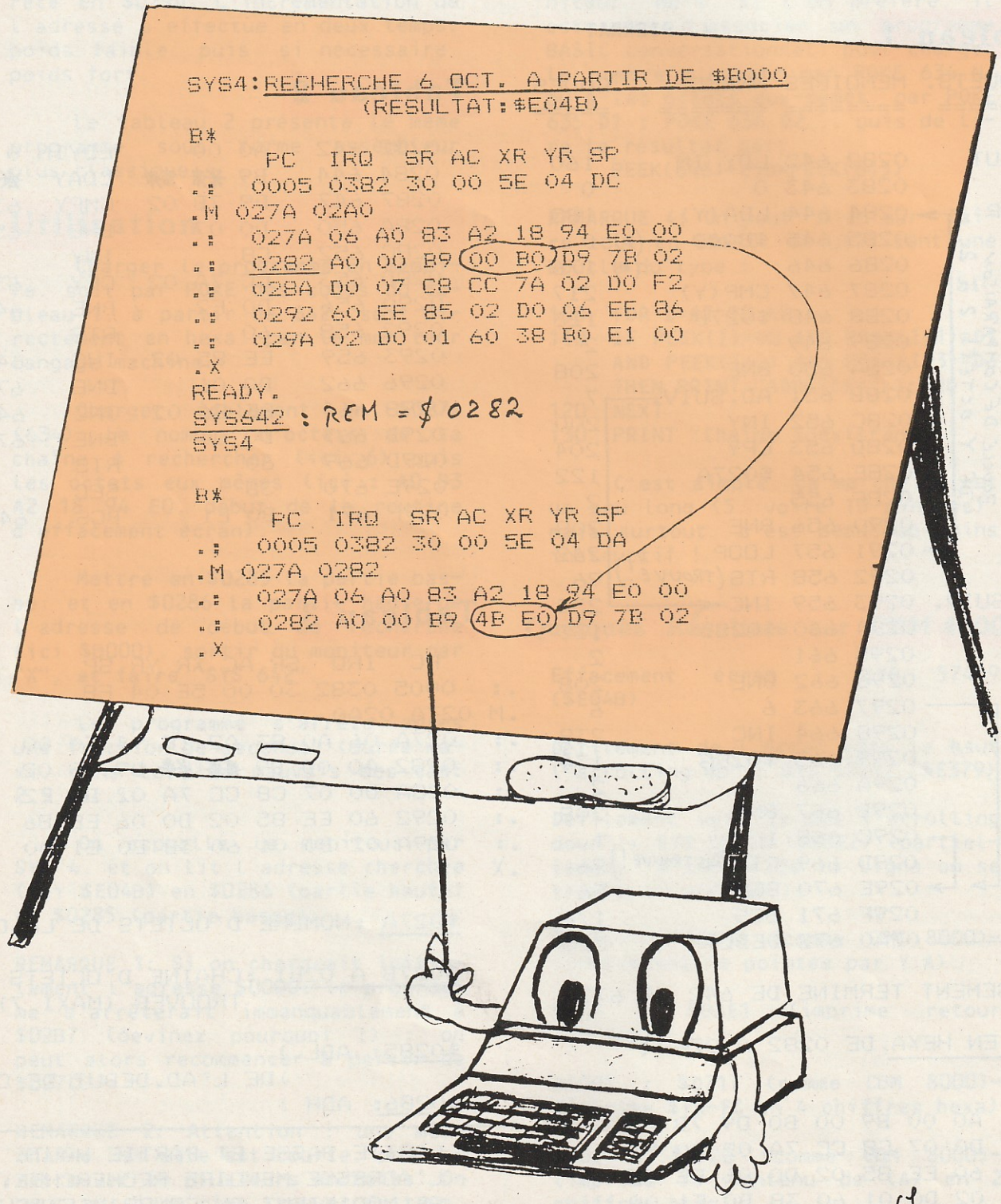
$0285: ADL)
)DE L'AD.DEBUT DE CHAINE
$0286: ADH)

(*) PARTIE BASSE ET PARTIE HAUTE DE L'ADRESSE MEMOIRE RECHERCHEE: SE MODIFIENT EN COURS D'EXECUTION.

Exemple

```
SYS4:RECHERCHE 6 OCT. A PARTIR DE $B000
          (RESULTAT:$E04B)
B*
     PC   IRQ   SR AC XR YR SP
.:   0005 0382 30 00 5E 04 DC
.M 027A 02A0
.:   027A 06 A0 83 A2 18 94 E0 00
.:   0282 A0 00 B9 00 B0 D9 7B 02
.:   028A D0 07 C8 CC 7A 02 D0 F2
.:   0292 60 EE 85 02 D0 06 EE 86
.:   029A 02 D0 01 60 38 B0 E1 00
.X
READY.
SYS642 : REM = $ 0282
SYS4

B*
     PC   IRQ   SR AC XR YR SP
.:   0005 0382 30 00 5E 04 DA
.M 027A 0282
.:   027A 06 A0 83 A2 18 94 E0 00
.:   0282 A0 00 B9 4B E0 D9 7B 02
.X
```



BAM BAM BAM ....

Visualisez avec précision l'occupation de vos disques

Introduction

Ce programme vous permet de visualiser la BAM (Block Availability Map) qui représente l'image ou la carte de la disquette du point de vue occupation des Pistes/Secteurs.

Une fois que vous aurez entré ce programme essayez pour une meilleure compréhension de la technique d'Allocation du Dos de :

- sauver un programme ; visualisez la BAM,

- le "scratcher"; revisualisez la BAM et comparez les deux

Cette connaissance visuelle vous permettra pour des raisons diverses de protéger une zone Piste/Secteur par la commande "B-A". Essayez de l'appliquer et visualisez la BAM.

Le programme BAM

Le programme se décompose en deux parties :

a) une partie en Basic qui ouvre les fichiers nécessaires pour la lecture de la BAM,

b) l'autre est la partie en LM. Elle travaille en utilisant les routines du noyau. De ce fait le programme doit marcher pour toutes les générations de PET/CBM. Rappelons que nous désignons par noyau la table d'adresses système se trouvant en haut de mémoire et permettant l'exécution d'une routine précise à une adresse commune quelque soit le type de PET/CBM.

Sinon, cette partie lit le secteur entier puis analyse les octets de la BAM, l'imprime de même que le nom de la disquette, la version du Dos et l'identificateur.

Quelques explications sur le formatage des disquettes

D'abord le DOS (Disk Operating System) est en mémoire morte dans la 4040 ou 3040. Il est complètement indépendant de l'unité centrale. Il gère entièrement les fichiers et la disquette (c'est-à-dire : catalogue Allocation et chainage des pistes/secteurs).
Le programme ne se préoccupe que de l'allocation des Pistes/Secteurs, donc pour la suite de mon explication j'ignorerai le reste.

1 - ALLOCATION DES PISTES/SECTEURS SUR LES DIFFERENTS DOS.

PISTES	SECTEURS	
	2040-3040	4040
1-17	0-20	0-20
18-24	0-19	0-18
25-30	0-17	0-17
31-35	0-16	0-16

Pour information voici la table de la 8050 :

PISTES	SECTEURS
1-39	0-28
40-53	0-26
54-64	0-24
65-70	0-22

Cette organisation, bien qu'à première vue paraissant compliquée, est l'une des meilleures que je connaisse.

2 - BAM et titre de la disquette

Elle se trouve sur le secteur 0 piste 18. Quelle est sa structure? Voici le tableau explicatif:

Octets	2040 ou 3040	4040
0-1	chaînage P/S	
2-3	formatage du Dos	
	1	2A
4-143	BAM proprement dite 140 octets au total. Chacune de ces 35 pistes a 40 octets	
144-161	nom de la disquette	
162-163	ID de la disquette	
165-166	-	2A
171-255	non utilisé	

Pour chaque octet de la BAM chacun des 8 bits indique si le secteur est utilisé. Par exemple si le bit a une valeur nulle le secteur est occupé, sinon il est évidemment libre.

Voici un exemple pour les 8 premiers octets de la piste 18 secteur 0 :

chaînage		version		a	b	c	d
12	01	01	00	15	FF	FF	1F

a) Total des secteurs libres pour cette piste : 21.

b) $FF = 1 1 1 1 1 1 1 1
- secteur 0
- secteur 1
- secteur 2
- secteur 3
- secteur 4
- secteur 5
- secteur 6
- secteur 7

c) $FF = 1 1 1 1 1 1 1 1
- secteur 8
- secteur 9
- secteur 10
- secteur 11
- secteur 12
- secteur 13
- secteur 14
- secteur 15

c) $1F = 0 0 0 1 1 1 1 1
- secteur 16
- secteur 17
- secteur 18
- secteur 19
- secteur 20

Les 3 octets de gauche ne sont pas utilisés dans cet exemple.

Références

Programming the PET/CBM par Raeto COLLIN WEST.

Programmes Internes du PET/CBM par B. MICHEL. BCM

Clefs pour le PET/CBM, D.J. DAVID, éditions PSI

Compute's First Book of PET/CBM, Compute.

N.B. Le "Dump" de la mémoire à pu être effectué à l'aide de MICROMON qui m'a été envoyé directement par l'auteur : j'en profite par ces lignes pour le remercier. Sinon ce proogramme est du domaine public et ne peut être en aucun cas vendu de même que le programme de BAM.

Thierry HENTIC
Pascal MARCHANDEAU

END OF MAE PASS!

```
                0010 ;PU "@1:BAM.M0"
                0020 ;*******************************
                0030 ;* PROGRAMME DE LECTURE DE LA BAM
                0040 ;* FONCTIONNE EN BASIC 2.0
                0050 ;*                  ET  4.0
                0060 ;* VERIFIE AUSSI LE DOS DE LA
                0070 ;* DISQUETTE(3040/4040)
                0080 ;*******************************
                0090 ;
                0100 ;
                0110            .LS
                0120            .OS
                0130            .BA $0500
                0140 ;
                0150 ;
                0160 SETIN      .DE $FFC6          ;ETABLIT UN PERIPH EN ENTR
                0170 WRT        .DE $FFD2          ;SORT UN CARACTERE
                0180 CLR        .DE $FFCC          ;RAZ DES SORTIES
                0190 GETIN      .DE $FFCF          ;LIT UN CARACTERE
                0200 PLEIN      .DE $51            ;ROND PLEIN
                0210 VIDE       .DE $57            ;ROND VIDE
0500- A2 02     0220            LDX #$02
0502- 20 C6 FF  0230            JSR SETIN          ;CHOISIT LE FICHIER #02 EN
0505- A0 00     0240            LDY #$00
0507- 20 CF FF  0250 LOOP       JSR GETIN          ;LECTURE DE LA TABLE
050A- 99 52 06  0260            STA BAM,Y
050D- C8        0270            INY
050E- D0 F7     0280            BNE LOOP           ;PAS FINI ??
0510- 20 CC FF  0290            JSR CLR            ;DESELECTION PERIPHERIQUE
0513- AD 54 06  0300 TEST       LDA BAM+2
0516- C9 01     0310            CMP #$01           ;CHERCHE LA VERSION DOS
0518- D0 06     0320            BNE TEST1
051A- 20 88 05  0330            JSR D3040          ;DOS 2 ->INIT TABLE
051D- 4C 27 05  0340            JMP FINTEST        ;RETOUR
0520- C9 41     0350 TEST1      CMP #$41
0522- D0 70     0360            BNE TESTERR        ;NON ERREUR ET MESSAGE
0524- 20 8E 05  0370            JSR D4040          ;DOS 4->INIT TABLE
0527- 20 DE 07  0380 FINTEST    JSR AFFIMESS       ;IMPRESSION MESSAGE
052A- 20 B2 05  0390            JSR BAMINIT
052D- AD 25 06  0400            LDA TABLE          ;INIT NBRE DE
0530- 8D 52 07  0410            STA INDICEX        ;SECTEUR ET PISTE
0533- AD 26 06  0420            LDA TABLE+1        ;A AFFICHER
0536- 8D 54 07  0430            STA INDY           ;PAR BAMPRINT
0539- 8D 53 07  0440            STA INDICEY        ;
053C- 20 6C 07  0450            JSR BAMPRINT       ;'BAMPRINT'
053F- AD 27 06  0460            LDA TABLE+2
0542- 8D 52 07  0470            STA INDICEX
0545- AD 28 06  0480            LA TABLE+3
0548- 8D 54 07  0490            STA INDY
054B- 8D 53 07  0500            STA INDICEY
054E- 20 6C 07  0510            JSR BAMPRINT       ;IDEM
0551- AD 29 06  0520            LDA TABLE+4
0554- 8D 52 07  0530            STA INDICEX
0557- AD 29 06  0540            LDA TABLE+5
055A- 8D 54 07  0550            STA INDY
055D- 8D 53 07  0560            STA INDICEY
0560- 20 6C 07  0570            JSR BAMPRINT       ;IDEM
0563- AD 2B 06  0580            LDA TABLE+6
0566- 8D 52 07  0590            STA INDICEX
0569- AD 2C 06  0600            LDA TABLE+7
056C- 8D 54 07  0610            STA INDY
056F- 8D 53 07  0620            STA INDICEY
0572- 20 6C 07  0630            JSR BAMPRINT       ;IDEM
                0640 ;*****************************
                0650 ;* ECRITURE NOM DE LA DISQUETTE
                0660 ;*****************************
                0670 ;
0575- A9 13     0680            LDA #$13           ;CURSEUR EN HAUT
0577- 20 D2 FF  0690            JSR WRT            ;ET A GAUCHE DE L'ECRAN
057A- A2 00     0700            LDX #$00
057C- BD E2 06  0710 FINNOM1    LDA BAM+144,X      ;XEME CARACTERE DU NOM
057F- F0 06     0720            BEQ FINNOM2        ;DE LA DISQUETTE
0581- 20 D2 FF  0730            JSR WRT
0584- E8        0740            INX                ;X=X+1
0585- D0 F5     0750            BNE FINNOM1
0587- 60        0760 FINNOM2    RTS
0588- A9 14     0770 D3040      LDA #$14           ;C'EST UNE 3040
058A- 8D 28 06  0780            STA TABLE+3        ;20 SECTEURS SUR PISTE 18
058D- 60        0790            RTS
058E- A9 13     0800 D4040      LDA #$13           ;C'EST UNE 4040
0590- 8D 28 06  0810            STA TABLE+3
0593- 60        0820            RTS
0594- A0 00     0830 TESTERR    LDY #$00
0596- B9 2E 06  0840            LDA ERREUR,Y
```

```
0599- F0 06      0850              BEQ FINTESTERR
059B- 20 D2 FF   0860              JSR WRT                 ;AFFICHAGE 'DISQUE NON REC
059E- C8         0870              INY
059F- D0 F5      0880              BNE TESTERR+2
05A1- 60         0890 FINTESTERR RTS
05A2- A0 00      0900 PRINTCHAR    LDY #$00
05A4- 91 00      0910              STA ($00),Y
05A6- 18         0920 PRINTCHAR1 CLC
05A7- A5 00      0930              LDA *$00
05A9- 69 28      0940              ADC #$28
05AB- 85 00      0950              STA *$00
05AD- 90 02      0960              BCC SKIP
05AF- E6 01      0970              INC *$01                ;LIGNE SUIVANTE
05B1- 60         0980 SKIP         RTS
                 0990 ;*******************************
                 1000 ;* INITIALISATION DES PARAMETRES
                 1010 ;*******************************
05B2- A9 A3      1020 BAMINIT      LDA #$A3
05B4- 8D 47 06   1030              STA POINTTABL
05B7- 85 00      1040              STA *$00                ;POINTEUR BAS DE LA BAM
05B9- A9 80      1050              LDA #$80
05BB- 8D 48 06   1060              STA POINTTABL+1 ;
05BE- 85 01      1070              STA *$01                ;PONTEUR HAUT DE LA BAM
05C0- A9 05      1080              LDA #$05                ;LA BAM COMMENCE AU 5EME
05C2- 8D 51 06   1090              STA ORGBYTEBAM          ;OCTET DE LA
05C5- A9 00      1100              LDA #$00                ;PISTE 18 SECTEUR 0
05C7- 8D 44 06   1110              STA SECTEUR             ;INIT NOMBRE
05CA- 8D 45 06   1120              STA SECTEUR+1           ;SECTEUR LIBRE
05CD- 8D 46 06   1130              STA BIT
05D0- 60         1140              RTS
05D1- 93 0D 0D   1150 MESSAGE2     .BY 147 $0D $0D $12 '   PISTES   1          2
05D4- 12 20 20
05D7- 20 50 49
05DA- 53 54 45
05DD- 53 20 20
05E0- 20 31 20
05E3- 20 20 20
05E6- 20 20 20
05E9- 20 20 32
05EC- 20 20 20
05EF- 20 20 20
05F2- 20 20 20
05F5- 33 20 20
05F8- 20 20 20
05FB- 0D
05FC- 12 20 20   1160              .BY $12 '   123456789012345678901234567890123 45
05FF- 20 31 32
0602- 33 34 35
0605- 36 37 38
0608- 39 30 31
060B- 32 33 34
060E- 35 36 37
0611- 38 39 30
0614- 31 32 33
0617- 34 35 36
061A- 37 38 39
061D- 30 31 32
0620- 33 34 35
0623- 0D 00
0625- 11 15 07   1170 TABLE        .BY 17 21 7 19 6 18 5 17 0
0628- 13 06 12
062B- 05 11 00
062E- 93 12 44   1180 ERREUR       .BY 147 18 'DISQUE NON RECONNU' 13 0
0631- 49 53 51
0634- 55 45 20
0637- 4E 4F 4E
063A- 20 52 45
063D- 43 4F 4E
0640- 4E 55 0D
0643- 00
0644-            1190 SECTEUR      .DS 2                   ;COMPTEUR SECTEUR LIBRE
0646-            1200 BIT          .DS 1 ;
0647-            1210 POINTTABL    .DS 2
0649-            1220 OCTET        .DS 8                   ;TRANSFORMATION DE L'OCTET
0651-            1230 ORGBYTEBAM   .DS 1                   ;INDICE DE L'OCTET DE LA
0652-            1240 BAM          .DS 256                 ;CARTE DE LA BAM
0752-            1250 INDICEX      .DS 1
0753-            1260 INDICEY      .DS 1
0754-            1270 INDY         .DS 1                   ;REG TEMP PISTE
0755- 53 45 43   1280 COLONNE      .BY 'SECTEURS' $60 $60 $60 $60 $60 $60 $60 $60
0758- 54 45 55
075B- 52 53 60
075E- 60 60 60
0761- 60 60 60
0764- 60 60
```

```
0766- 60 60 60  1290              .BY $60 $60 $60 $60
0769- 60
076A-           1300 SECTEUR.CO .DS 2
                1310 ;*************************
                1320 ;IMPRESSION DE LA CARTE
                1330 ;*************************
076C- AE 51 06  1340 BAMPRINT    LDX ORGBYTEBAM
076F- BD 52 06  1350             LDA BAM,X
0772- A2 07     1360             LDX #$07
0774- 0A        1370 BAMPR1      ASL A                ;TEST LE BIT + A GAUCHE
0775- 90 0A     1380             BCC ZERO             ;SECTEUR LIBRE??
0777- 48        1390             PHA                  ;OUI
0778- A9 57     1400             LDA #VIDE            ;SECTEUR LIBRE
077A- 9D 49 06  1410             STA OCTET,X
077D- 68        1420             PLA
077E- 4C 90 07  1430             JMP DECREMENT
0781- 48        1440 ZERO        PHA
0782- A9 51     1450             LDA #PLEIN           ;SECTEUR OCCUPE
0784- 9D 49 06  1460             STA OCTET,X
0787- EE 44 06  1470             INC SECTEUR
078A- D0 03     1480             BNE DECREMENT-1
078C- EE 45 06  1490             INC SECTEUR+1
078F- 68        1500             PLA
0790- CA        1510 DECREMENT   DEX
0791- 10 E1     1520             BPL BAMPR1
0793- AE 46 06  1530 AFFOCTET    LDX BIT
0796- BD 49 06  1540             LDA OCTET,X
0799- 20 A2 05  1550             JSR PRINTCHAR
079C- EE 46 06  1560             INC BIT
079F- AD 46 06  1570             LDA BIT
07A2- C9 08     1580             CMP #$08
07A4- D0 08     1590             BNE AFFOCTET1
07A6- A9 00     1600             LDA #$00
07A8- 8D 46 06  1610             STA BIT
07AB- EE 51 06  1620             INC ORGBYTEBAM
07AE- CE 53 07  1630 AFFOCTET1   DEC INDICEY
07B1- F0 02     1640             BEQ CHANGEPIST
07B3- D0 B7     1650             BNE BAMPRINT
07B5- AD 54 07  1660 CHANGEPIST  LDA INDY             ;CHGT DE PISTE
07B8- 8D 53 07  1670             STA INDICEY
07BB- A9 00     1680             LDA #$00
07BD- 8D 46 06  1690             STA BIT
07C0- EE 51 06  1700             INC ORGBYTEBAM
07C3- EE 51 06  1710             INC ORGBYTEBAM
07C6- EE 47 06  1720             INC POINTTABL
07C9- D0 03     1730             BNE SKIP1
07CB- EE 48 06  1740             INC POINTTABL+1
07CE- AD 47 06  1750 SKIP1       LDA POINTTABL
07D1- 85 00     1760             STA *$00
07D3- AD 48 06  1770             LDA POINTTABL+1
07D6- 85 01     1780             STA *$01
07D8- CE 52 07  1790             DEC INDICEX
07DB- D0 8F     1800             BNE BAMPRINT
07DD- 60        1810 RETOUR      RTS
                1820 ;*************************
                1830 ;*IMPRESSION DES MESSAGES
                1840 ;*************************
07DE- A2 00     1850 AFFIMESS    LDX #$00             ;AFFICHE
07E0- BD D1 05  1860 MSG1        LDA MESSAGE2,X       ;MESSAGE 'PISTE
07E3- F0 06     1870             BEQ MESFIN2          ;
07E5- 20 D2 FF  1880             JSR WRT              ;
07E8- E8        1890             INX                  ;
07E9- D0 F5     1900             BNE MSG1             ;
07EB- A2 AF     1910 MESFIN2     LDX #$AF             ;INIT NB SECTEURS-UNITES
07ED- 8E 6A 07  1920             STX SECTEUR.CO       ;A  01 EN ASCII ET RVS
07F0- E8        1930             INX                  ;
07F1- 8E 6B 07  1940             STX SECTEUR.CO+1     ;INIT NB SECTEURS-DIZ.
07F4- A9 A0     1950             LDA #$A0             ;INIT PTRS PRINTCHAR1
07F6- 85 00     1960             STA *$00             ;A $80A0 QUI EST L'ADRESSE
07F8- A9 80     1970             LDA #$80             ;DE DEBUT DE LA
07FA- 85 01     1980             STA *$01             ;5 IEME LIGNE
07FC- A2 00     1990             LDX #$00             ;SERT DE PTR POUR LE TEXTE
07FE- A0 00     2000 COL         LDY #$00             ;POINTE 1 ERE COL.
0800- BD 55 07  2010             LDA COLONNE,X        ;A←CARACTERE A AFFICHER
0803- 18        2020             CLC
0804- 69 40     2030             ADC #$40             ;PASSE A EN REVERSE
0806- 91 00     2040             STA ($00),Y          ;AFFICHE A
0808- C8        2050             INY                  ;OFFSET = 1
0809- EE 6A 07  2060             INC SECTEUR.CO       ;INCREM. COMPT. UNITE
080C- AD 6A 07  2070             LDA SECTEUR.CO       ;REGARDE SI > 9
080F- C9 BA     2080             CMP #$BA
0811- D0 0F     2090             BNE SUITE            ;NON SAUTE
0813- EE 6B 07  2100             INC SECTEUR.CO+1     ;OUI INCREM. DIZ.
0816- AD 6B 07  2110             LDA SECTEUR.CO+1     ;AFFICHE DIZ.
```

```
0819- 91 00      2120              STA ($00),Y
081B- A9 B0      2130              LDA #$B0                  ;UNITE = 0
081D- 8D 6A 07   2140              STA SECTEUR.CO
0820- D0 04      2150              BNE SUITE1                ;FAIT TOUJOURS UN SAUT
0822- A9 A0      2160 SUITE        LDA #$A0                  ;AFFICHAGE ESPACE EN REVER
0824- 91 00      2170              STA ($00),Y
0826- AD 6A 07   2180 SUITE1       LDA SECTEUR.CO            ;AFFICHAGE UNITE
0829- C8         2190              INY
082A- 91 00      2200              STA ($00),Y
082C- E8         2210              INX
082D- E0 15      2220              CPX #$15                  ;21 SECTEURS ??
082F- F0 06      2230              BEQ FINCOL
0831- 20 A6 05   2240              JSR PRINTCHAR1
0834- 4C FE 07   2250              JMP COL
0837- 60         2260 FINCOL       RTS
                 2270              .EN
```

END OF THE PASS!

```
AFFIMESS =07DE          AFFOCTET =0793          AFFOCTET1 =07AE
BAM =0652               BAMINIT =05B2           BAMPR1 =0774
BAMPRINT =076C          BIT =0646               CHANGEPIST =07B5
CLR =FFCC               COL =07FE               COLONNE =0755
D3040 =0588             D4040 =058E             DECREMENT =0790
ERREUR =062E            FINCOL =0837            FINNOM1 =057C
FINNOM2 =0587           FINTEST =0527           FINTESTERR =05A1
GETIN =FFCF             INDICEX =0752           INDICEY =0753
INDY =0754              LOOP =0507              MESFIN2 =07EB
MESSAGE2 =05D1          MSG1 =07E0              OCTET =0649
ORGBYTEBAM =0651        PLEIN =0051             POINTTABL =0647
PRINTCHAR =05A2         PRINTCHAR1 =05A6        RETOUR =07DD
SECTEUR =0644           SECTEUR.CO =076A        SETIN =FFC6
SKIP =05B1              SKIP1 =07CE             SUITE =0822
SUITE1 =0826            TABLE =0625             TEST =0513
TEST1 =0520             TESTERR =0594           VIDE =0057
WRT =FFD2               ZERO =0781
```

```
10 CLOSE2:CLOSE15
12 INPUT"COUEL DRIVE ■■■1■■■■■";D$
20 OPEN15,8,15
30 PRINT#15,"I0"
40 OPEN2,8,2,"#"
50 PRINT#15,"U1:";2;D$;18;0
60 PRINT#15,"B-P";2;0
70 SYS1280
73 PRINT#15,"I";D$
80 CLOSE2:CLOSE15
90 WAIT152,1
100 END
```

```
0500  A2 02 20 C6 FF A0 00 20 CF FF 99 52 06 C8 D0 F7   ". F? . O?.R.HP7
0510  20 CC FF AD 54 06 C9 01 D0 06 20 88 05 4C 27 05   L?-T.I.P. ..L'.
0520  C9 41 D0 70 20 8E 05 20 DE 07 20 B2 05 AD 25 06   IAP0 .. ↑. 2.-%.
0530  8D 52 07 AD 26 06 8D 54 07 8D 53 07 20 6C 07 AD   .R.-&..T..S. ,.-
0540  27 06 8D 52 07 AD 28 06 8D 54 07 8D 53 07 20 6C   '..R.-(..T..S. ,
0550  07 AD 29 06 8D 52 07 AD 2A 06 8D 54 07 8D 53 07   .-)..R.-*..T..S.
0560  20 6C 07 AD 2B 06 8D 52 07 AD 2C 06 8D 54 07 8D    ,.-+..R.-,..T..
0570  53 07 20 6C 07 A9 13 20 D2 FF A2 00 BD E2 06 F0   S. ,.). R?".=".0
0580  06 20 D2 FF E8 D0 F5 60 A9 14 8D 28 06 60 A9 13   . R?(P5 )..(. ).
0590  8D 28 06 60 A0 00 B9 2E 06 F0 06 20 D2 FF C8 D0   .(.  .9..0. R?HP
05A0  F5 60 A0 00 91 00 18 A5 00 69 28 85 00 90 02 E6   5  ....%.)(....&
05B0  01 60 A9 A3 8D 47 06 85 00 A9 80 8D 48 06 85 01   . )#.G...)..H...
05C0  A9 05 8D 51 06 A9 00 8D 44 06 8D 45 06 8D 46 06   )..Q.)..D..E..F.
05D0  60 93 0D 0D 12 20 20 20 50 49 53 54 45 53 20 20    ....    PISTES
05E0  20 31 20 20 20 20 20 20 20 20 20 20 32 20 20 20    1          2
05F0  20 20 20 20 20 20 33 20 20 20 20 20 0D 12 20 20        3       ..
0600  31 32 33 34 35 36 37 38 39 30 31 32 33 34 35 36   1234567890123456
0610  37 38 39 30 31 32 33 34 35 36 37 38 39 30 31 32   7890123456789012
0620  33 34 35 0D 00 11 15 07 13 06 12 05 11 00 93 12   345.............
0630  44 49 53 51 55 45 20 4E 4F 4E 20 52 45 43 4F 4E   DISQUE NON RECON
0640  4E 55 0D 00 00 00 00 00 00 00 00 00 00 00 00 00   NU..............
0650  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0660  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0670  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0680  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0690  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
06A0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
06B0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
06C0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
06D0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
06E0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
06F0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0700  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0710  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0720  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0730  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0740  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0750  00 00 00 00 00 53 45 43 54 45 55 52 53 60 60 60   .....SECTEURS
0760  60 60 60 60 60 60 60 60 60 60 00 00 AE 51 06 BD               ...Q.=
0770  52 06 A2 07 0A 90 0A 48 A9 57 9D 49 06 68 4C 90   R."....H)W.I.(L.
0780  07 48 A9 51 9D 49 06 EE 44 06 D0 03 EE 45 06 68   .H)Q.I..D.P..E.(
0790  CA 10 E1 AE 46 06 BD 49 06 20 A2 05 EE 46 06 AD   J.!.F.=I. "..F.-
07A0  46 06 C9 08 D0 08 A9 00 8D 46 06 EE 51 06 CE 53   F.I.P.)..F..Q.NS
07B0  07 F0 02 D0 B7 AD 54 07 8D 53 07 A9 00 8D 46 06   .0.P7-T..S.)..F.
07C0  EE 51 06 EE 51 06 EE 47 06 D0 03 EE 48 06 AD 47   .Q..Q..G.P..H.-G
07D0  06 85 00 AD 48 06 85 01 CE 52 07 D0 8F 60 A2 00   ...-H...NR.P. ".
07E0  BD D1 05 F0 06 20 D2 FF E8 D0 F5 A2 AF 8E 6A 07   =Q.0. R?(P5"/.*.
07F0  E8 8E 6B 07 A9 A0 85 00 A9 80 85 01 A2 00 A0 00   (.+.) ..)...". .
0800  BD 55 07 18 69 40 91 00 C8 EE 6A 07 AD 6A 07 C9   =U..)@..H.*.-*.I
0810  BA D0 0F EE 6B 07 AD 6B 07 91 00 A9 B0 8D 6A 07   :P..+.-+...)0.*.
0820  D0 04 A9 A0 91 00 AD 6A 07 C8 91 00 E8 E0 15 F0   P.) ..-*.H..( .0
0830  06 20 A6 05 4C FE 07 60 00 AA AA AA AA AA AA AA   . &.L>. .*******
```

JANAL

Votre équipe Rhône-Alpes

commodore

VENEZ VOUS INITIER A LA MICRO INFORMATIQUE
POUR VOTRE UTILISATION PERSONNELLE

VENEZ ESSAYER VOUS-MEME
LES DEVELOPPEMENTS "JANAL" SUR LE MATERIEL "COMMODORE"

JANAL *Lyon*	1, Place Chazette 69001 Lyon Tél. (7) 839.44.76	S.A.V. 12, Crs d'Herbouville 69004 Lyon Tél. (7) 839.77.02
JANAL *Grenoble*		9, Quai Claude Bernard 38000 Grenoble Tél. (76) 43.10.65
JANAL *St Etienne*		1, Rue Badouillère 42100 Saint-Etienne Tél. (77) 38.48.55
JANAL *Savoies*	12, Rue de la Paix 74000 Annecy Tél. (50) 45.24.27	2 bis, Route d'Annecy 74150 Rumilly Tél. (50) 01.42.56
JANAL *Automatisme*		REP 6, rue Docteur Vacher 69720 St-Laurent-de-Mûre Tél. (7) 840.90.33

Un CBM secrétaire de laboratoire

Une utilisation dans le domaine médical, mais le module de saisie des données sur écran peut s'appliquer à d'autres domaines.

La croissance rapide de la demande d'examens de laboratoire faisait craindre une lente, mais certaine, asphyxie du secrétariat. De plus, la présentation des documents rédigés à la main laissait à désirer.

Tentés par les techniques de l'informatique, nous avions envisagé d'installer un mini-ordinateur.

Une enquête menée en France et en Allemagne, nous a fait renoncé à ce projet. Les inconvénients principaux étant :

- le côut très élevé du matériel
- la relative lourdeur des systèmes
- la nécessité de la présence permanente d'au moins un informaticien
- le coût élevé de la gestion.

L'apparition sur le marché en 1978 des premiers micro-ordinateurs et leur très faible coût nous a amenés à reprendre notre projet.

Pour éviter de perturber le fonctionnement du secrétariat et laisser s'habituer le personnel, nous avons choisi une philosophie de prudence et de progressivité. Pour cela, nous avons sélectionné un problème bien défini, marginal par rapport au fonctionnement journalier. Qui de plus par son incidence financière amortirait rapidement le coût du matériel.

Le premier programme

La facturation des examens de laboratoire pour certains malades sortants répondait à nos critères pour un premier essai. Il s'agissait d'éditer un document détaillant les examens avec leur nombre et le montant total. Ce travail pouvant s'effectuer à des heures creuses.

Je l'ai rédigé, avec comme objectif la simplicité et la rapidité de la saisie. Chaque examen étant codé sur deux caractères (une lettre, un chiffre).

Le nombre d'examens était frappé dans une boucle de GET.

Le document obtenu par sa clarté a permis d'améliorer très fortement le recouvrement des sommes dues.

Ce premier succès nous a conduit à viser un deuxième objectif plus ambitieux.

LA SAISIE journalière des examens de laboratoires demandés par les divers services de l'hôpital, le personnel, les hopitaux extérieurs.

Pour cela, nous avons amélioré le matériel.

Le PET 8 K a été remplacé par un CBM 32 K.
Le magnétophone par une unité de double disquette.
L'imprimante CENTRONICS à rouleau par une petite CBM 80 colonnes.

Les buts du nouveau programme

Saisie de l'identité du malade
Saisie du service selon une lettre ou 1 chiffre clef (A à Z, 1 à 2)
Saisie des examens ordinaires demandés.
Eventuellement, saisie d'examens peu habituels et des prélèvements.

Copie des saisies sur disquette avec numérotation automatique des malades successifs.
Exploitation de la saisie pour éditer le journal légal et la facturation des examens.

Les impératifs du programme

- la rapidité de la saisie
- la facilité d'emploi
- la clarté de la visualisation
- la sécurité de la saisie
- la rapidité de l'exploitation et de l'édition.

Les principaux points du programme

Un menu permet de choisir entre
- Débuter le programme de saisie journalier
- Reprendre une saisie interrompue (panne d'électricité, etc.)
- Lire le fichier de regroupement par service
- Terminer les opérations.

Le début de saisie demande le jour (mois et année sont mis automatiquement).

La saisie d'identité

Une technique spéciale d'INPUT est utilisée. On ouvre une file sur le clavier considéré comme périphérique extérieur. Le clavier comporte un tampon de dix caractères qui se vide automatiquement dans le tampon de saisie du Basic.
Ceci permet une présentation beaucoup plus agréable, toutes les questions étant affichées simultanément à l'écran.
Le mécanisme est très simple, le curseur étant envoyé successivement sur les lignes à saisir. (lignes 625 à 808 du programme).

La saisie terminée, on passe à...

La saisie des examens

Après la saisie d'identité, le tableau des examens est affiché à l'écran.
Il comprend 8 colonnes et 19 lignes.
Les examens sont affichés sur quatre caractères, ce qui oblige à des abréviations. Celles-ci sont choisies pour être claires.
Un curseur clignotant (lignes 1860-1900) marque l'examen à saisir. Il se déplace par les curseurs habituels, un à droite saute une colonne, un en bas saute une ligne, par exemple.
De plus le tableau numérique considéré comme un miroir de l'écran, permet des déplacements rapides.
Par exemple le 5 envoie au centre, le 9 en haut à droite etc.
Les lignes 1630 à 1770 assurent cette gestion d'écran par des GET.
La touche "ESPACE" ou la touche " * " permettent la saisie d'un examen. Ce qui se manifeste à l'écran par une inversion vidéo.
On peut toujours annuler une saisie en revenant sur l'examen et frappant la touche" - " arithmétique.
La saisie a aussi pour effet de coder l'examen et de l'ajouter dans la chaîne de caractère LY$.
Le code consiste à remplacer le numéro d'examen par un code ASCII obtenu en évitant les caractères qui seraient gênants à la saisie sur disquette tels que chr$(0) ou chr$(13).

Fin de la saisie sur le tableau

On l'obtient, soit que l'on ait atteint le nombre limite d'examens choisis, soit le plus souvent en frappant la touche "OFF".
On demande alors si toute l'ordonnance est saisie. Sinon, on appelle un deuxième tableau de saisie d'examens plus rares et même la possibilité de saisir un examen

en clair.
On passe ensuite à la saisie éventuelle des prélèvements.

Saisie des prélèvements

Celle-ci est présentée en tableau selon une liste correspondant à la classification de la Sécurité Sociale.

A chaque prélèvement, on demande la qualification du préleveur.
Il y a refus, si celle-ci n'est pas suffisante.

Ordonnance facturable ou non

Certains examens ne peuvent être facturés qu'après retour des résultats du laboratoire. C'est pratiquement la règle en bactériologie.
Cela est noté par code spécial
c pour facturable
i pour facturation différée.
Le dernier renseignement déclenche la copie sur disquette de la saisie.

Saisie sur disquette

Pour des raisons de simplicité, de clarté et pour éviter bien des erreurs, nous avons choisi la création de fichiers séquentiels.
Comme le nombre d'ordonnances se situe entre 100 et 140 par jour, nous avons voulu profiter de la commodité du directory et de la puissance du traitement de chaînes disponible sur le CBM.
Pour cela, un nom de fichier est attribué à chaque patient. Le nom est construit à partir d'éléments essentiels de la saisie. Il s'obtient par concaténation
- de la lettre clef du service (de A à Z puis de 1 à 2)
- du code de facturation (c ou i)
- du numéro sur trois chiffres du patient.

On concatène à gauche le numéro de la platine et à droite la chaîne
",S,W" pour l'écriture
",S,R" pour la lecture
De plus, un autre fichier général concatène en les regroupant dans une chaîne pour chaque service, les saisies successives.

Cette méthode présente de très nombreux avantages :
- en éditant le directory on a une image selon une clef choisie, des ordonnances.
- L'exploitation se fait avec une excellente rapidité, la recherche de chaque fichier étant très simple.
- On travaille avec une sécurité beaucoup plus grande qu'en accès direct.
- Selon les besoins d'exploitation, on peut étendre le nom de chaque fichier à seize caractères. Ce qui donne d'énormes possibilités d'exploitation selon des critères préalablement choisis.

Evidemment, comme toujours, il y a un revers la médaille, c'est la place perdue sur la disquette.
Pour nous, cela n'est pas gênant. En effet, la capacité relativement faible des disquettes ne nous permet pas de gérer notre fichier d'environ 10000 malades. Nous sommes donc amenés à garder notre technique des fiches cumulatives obtenues par photocopie. De ce fait, qu'importe de remplir plus ou moins la disquette. Il ne faut jamais chercher à faire entrer un éléphant dans une 2 CV.
Pour notre part, nous utilisons une disquette par jour de la semaine. Un programme "EFFACE" remet automatiquement les disquettes à blanc après la fin de l'exploitation.
Nous pensons bientôt relier notre CBM à une mémoire à disque type Winchester et bien sûr alors, nous changerons notre façon d'opérer.

Edition du résultat

En début d'après-midi, nous appelons la routine IMPRESSION LISTES SERVICES qui affiche à l'écran les noms des services ayant envoyé des examens. Nous répondons aux questions demandant le 1er et le dernier service à éditer. En général, la totalité est demandée.
Le programme va lire alors le

fichier général, en extrait les chaînes correspondant à chaque service, les découpe en fichiers malades qui sont lus et édités à raison de quatre par page.

Pour chaque service, on saute à la page suivante.

L'impression obtenue avec la petite imrimante CBM 3022 est superbe et permet d'obtenir des documents parfaits faciles à archiver.

Le document comporte tous les éléments essentiels concernant l'ordonnance.

- Identité du malade (Nom, prénom, âge, sexe, numéro d'hospitalisation)
- Service émetteur
- Numéro d'ordre légal
- Examens demandés
- Facturation

Reprise d'une saisie

Si pour une raison quelconque (panne d'électricité, erreur, etc.) la saisie a été interrompue, une routine "Reprise saisie" permet de reprendre la saisie après le dernier enregistrement sans perte d'information. Elle reprend entre autres, sans que l'on ait à intervenir, la numérotation des malades.

Conclusion

Le micro-ordinateur CBM 32 K avec une unité de disquette 3040 et une petite imprimante nous a permis d'apporter une grande amélioration au fonctionnement d'un important laboratoire central hospitalier.

Financièrement, il a permis le recouvrement de sommes importantes par l'édition claire et rapide des frais d'examens concernant certains malades sortants.

Il a permis une saisie rapide des examens journaliers et une très belle édition du journal légal et de la facturation.

Il n'y a pas eu de problèmes notables d'introduction de ces méthodes vis-à-vis du personnel, mais au contraire émulation.

Le personnel utilisé a été choisi parmi les infirmières et infirmiers et travaille sans difficulté en dehors de la présence de tout informaticien, avec un taux d'erreur très faible.

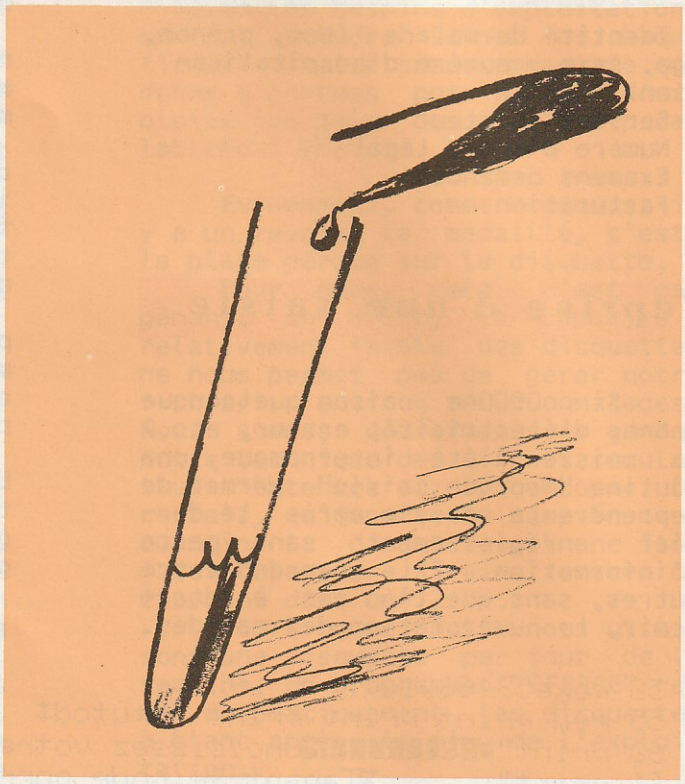
Je pense que notre façon très progressive de développer l'information du secrétariat a grandement contribué à ces résultats très positifs.

Ceci nous encourage à passer bientôt à des projets plus ambitieux comportant entre autres la gestion des résultats et du fichier général des malades.

Jean DELAVILLE

```
50 poke59468,14
55 rem print:print:print:inputod$
60 print"■■■■■■■■■■Choisissez votre action en■"
61 print"        ■Tapant le code correspondant■"
62 print"        ■et ATTENDEZ■"
63 fori=1to1000:next
64 print"■■                    ■MENU■"
70 print:print:print"■T■erminer SAISIE■"
71 print:print:print"■D■ebut SAISIE
72 print:print:print"■R■eprise SAISIE
73 print:print:print"■L■ecture FICHIER GENERAL"
74 print:print:print"■I■mpression LISTES/SERVICES"
77 print"■■TAPEZ LE CODE VOULU indique"
78 print"par la LETTRE en ■INVERSE■"
79 print"■■■■■■Et ATTENDEZ■":poke158,0
80 getz$
81 ifz$="d"thengosub100:goto580
82 ifz$="r"thenclr:gosub100:gosub5500:goto600
83 ifz$="l"thenclr:gosub100:goto5600
84 ifz$="i"thenclr:gosub100:goto5700
85 ifz$="t"then6000
90 goto80
91 end
```

```
100 d0$="0:":e0$="00:":d1$="1:":e1$="01:":rs$=",s,r":ws$=",s,w":c$=chr$(13)
110 dt$="80.":fs$="fs":k=0 :dimaa(28)
120 diml$(29):fori=1to29:l$(i)="":next
125 bd$="                    ":fori=1to3:bd$=bd$+bd$:next
130 br$="────────────────────────────────────────────────────────"
140 tt=20:dimli%(tt),co%(tt):poke144,46
150 open15,8,15
170 dimsv$(29),d1$(151),d2$(151):gosub2610
180 dimse$(20):se$(1)="phenobarbital":se$(2)="phenytoine":se$(3)="tegretol"
185 dimsf$(20):sf$(1)="100":sf$(2)="100":sf$(3)="100"
190 se$(4)="zarontin":se$(5)="mysoline":se$(6)="morphine":se$(7)="cocaine"
195 sf$(4)="100":sf$(5)="100":sf$(6)="100":sf$(7)="100"
200 se$(8)="amphetamines":se$(9)="barbituriques":se$(10)="methadone"
205 sf$(8)="100":sf$(9)="100":se$(10)="100 "
210 sv$(1)="SM5 Pav A"
220 sv$(2)="SM5 Pav B"
230 sv$(3)="SM18 Pav C"
240 sv$(4)="SM24 Pav D"
250 sv$(5)="ESQUIROL  "
260 sv$(6)="FERRUS    "
270 sv$(7)="SM24 Pav G"
280 sv$(8)="SM23 Pav H"
290 sv$(9)="SM23 Pav I"
300 sv$(10)="SM24 Pav J"
310 sv$(11)="SM24 Pav K"
320 sv$(12)="SM19 Pav L"
330 sv$(13)="MOREL    "
340 sv$(14)="P.JANET  "
350 sv$(15)="CHIRURGIE "
360 sv$(16)="PINEL    "
370 sv$(17)="Bq.du Sang"
380 sv$(18)="REANIMATION"
390 sv$(19)="STEREOTAXIE"
400 sv$(20)="NEUROCHIR. "
410 sv$(21)="SHU HOMMES"
420 sv$(22)="SHU FEMMES"
430 sv$(23)="NEUROLOGIE"
440 sv$(24)="EXTERIEURS"
450 sv$(25)="Hop.Exter."
460 sv$(26)="CABANIS  "
470 sv$(27)="PERSONNEL "
475 sv$(28)="Pichot"
477 sv$(29)="M.Prev."
480 bl$="                                        "
490 ma=0:rem marge
500 mj$=chr$(17):rem pour taper majuscules/minuscules
510 xc=8:rem nbre col
520 xl=19:rem maxi lig
530 tz=5:rem taille zone
540 sl=2:rem nbre saut lig
550 vz=4:rem taille cle
560 v=7:input"votre code";cd$
570 return
580 print"[redon][rvon]                    [rvof]";
590 input"date du jour[rvon] xx.08.81[rvof]";dj$:print"[redon]Attendez":gosub4760
    :print"[redon]"
600 no=no+1:in=0:t=0:b=0:ly$="":fi$="":fori=0tott:li%(i)=0:co%(i)=0:next
610 c1=0:c2=0:c3=0:t=0:fr=fr(o):ix=0
620 n0$=right$("00"+mid$(str$(no),2),3)
625 open5,0
```



```
630 dl$=dj$+n0$
640 gosub1330:print"▒nom ...............?
650 print"▒prenom ...........?
660 print"▒date de naissance ?
670 print"▒sexe ..............? f
680 print"▒No d'hospit.......? 80.xx.xxxx
690 print"▒date d'entree.....? jj.mm.81
700 fl=1:print"▒service ..........?
710 fl=0
720 print"▒▒▒▒▒";:gosub1280:no$=x$
730 gosub1280:pr$=x$
740 gosub1280:dn$=x$
750 gosub1280:sx$=x$
760 dp=0:gosub1280:iflen(x$)=10thenho$=x$:goto780
770 ho$=dt$+x$
780 dp=0:gosub1280:de$=x$
790 fl=1:gosub1280:md$=x$
800 gosub1570:ifz$="▒"orin=0then640
805 close5
806 print"▒Si Saisie finie Taper 'O',sinon 'N'▒"
807 getz$:ifz$="o"then830
808 ifz$<>"n"then807
810 gosub4180
820 gosub4130
830 print"▒▒le total de 'B' est il complet ?        "
840 print:printt:print:print"▒▒▒▒oui":print"▒▒▒▒non
850 print"▒▒▒▒▒▒Entrez le Code▒ et ▒Attendez"
860 forj=0toin-1:li%=li%(j):co%=co%(j):gosub1540:t=t+b:next:t$=mid$(str$(t),2)
870 getz$:ifz$="n"then910
880 ifz$<>"o"then870
890 tm$="c":fi$=a$+tm$:gosub1340
900 goto920
910 tm$="i":fi$=a$+tm$:gosub1340
920 gosub2330:ifl$(aa)=""thenl$(aa)=a$+tm$+n0$:goto930
925 l$(aa)=l$(aa)+tm$+n0$
930 cc$=l$(aa):mc$=left$(cc$,1):gosub4790
940 print"▒Relire >>>▒oui,▒non"
950 getz$:
960 ifz$="o"thenprint"▒▒▒▒▒▒▒▒Lecture de la Saisie":gosub1120:z$="n"
970 ifz$<>"n"then950
980 print"▒Impression >>>▒oui/▒non"
990 getz$:ifz$="o"thengosub4530:goto1010
1000 ifz$<>"n"then990
1010 print:print
1060 z$="" :print"Pour ▒Continuer▒ Tapez la▒ Barre▒,▒Sinon▒ votre ▒Code▒"
1070 getz$:ifz$=""then1070
1080 ifz$=" "then600
1090 ifz$<>left$(cd$,1)then600
1092 getz$:ifz$=""then1092
1095 ifz$<>mid$(cd$,2,1)then600
1100 goto60
1110 close15:end
1120 openv,8,v,d0$++fi$+n0$+rs$
1130 input#v,a0$,a1$,a2$,a3$,a4$,a5$,a6$,a7$,a8$,a9$,b1$,b2$,b3$
1140 a$=a5$:gosub2330:a5$=sv$(aa)
1150 printa2$" "a3$,a8$,a1$,a4$,a5$,a7$,"Tot.B"a9$:printa0$"  "b1$" "b2$"  "
1160 c1=val(b1$):c2=val(b2$):c3=val(b3$)                                  b3$
1170 ifc1>0thenprint"prelev.";:fori=1toc1:input#v,a1$(i):printa1$(i)" ";:next
```

```
1172 ifc1>0theninput#v,a$:print"prix="a$:s1=val(a$)
1180 s2=0:ifc2>0thenfori=1toc2:input#v,a2$(i):v2=(val(a2$(i))):printse$(v2)"  "
sf$(v2)
1182 ifc2>0thens2=s2+val(sf$(v2)):next
1190 ifc3>0thenfori=1toc3:input#v,a3$(i):printa3$(i):next:input#v,a$:print"prix
="a$
1200 input#v,a$:printa$
1210 forj=1tolen(a6$):ww=asc(mid$(a6$,j,1)):ifww>175andww<254thenx=ww-133:goto1
250
1220 ifww=254thenx=33:goto1250
1230 ifww=255thenx=12:goto1250
1240 x=ww-1
1250 gosub1550:printli$,b:nextj
1255 print"prix total="t;"+"s2;"+"ix;"="t+s2+ix
1260 closev:return
1270 end
1280 printtab(20+dp):input#5,x$:print:print:ifx$="?"andfl=1thengosub2050:fl=0
1290 iffl=1thenq$=x$:fl=0:gosub2330
1300 return
1310 gosub2050
1320 return
1330 printleft$("▒▒     le : "+dj$+"     numero :"+str$(no)+bl$,40):print:print:f
l=0:return
1340 u=6:openu,8,u,d0$+fi$+n0$+ws$:gosub1520
1350 in$=mid$(str$(in),2):t$=mid$(str$(t),2)
1360 c1$=mid$(str$(c1),2):c2$=mid$(str$(c2),2):c3$=mid$(str$(c3),2)
1370 print#u,in$c$dl$c$no$c$pr$c$sx$c$md$c$;
1380 fori=0toin-1:ww=8*li%(i)+co%(i)-8
1390 ifww=34thenww=254
1400 ifww=13thenww=255
1410 ifww=44orww=108orww=59orww=123orww=58orww=122thenww=ww+132
1420 ly$=ly$+chr$(ww):next:print#u,ly$c$;
1430 print#u,dn$c$;
1440 print#u,ho$c$;
1450 print#u,t$c$;
1460 print#u,c1$c$c2$c$c3$c$;
1470 ifc1>0thenfori=1toc1:print#u,ko$(i)c$;:next:print#u,px$c$;
1480 ifc2>0thenfori=1toc2:print#u,sa$(i)c$;:next
1490 ifc3>0thenfori=1toc3:print#u,ns$(i)c$;:next:print#u,ix$c$;
1500 print#u,tm$c$;
1510 closeu:return
1520 input#15,a,a$:ifathenprinta$:stop
1530 return
1540 x=li%*8+co%-9
1550 li$=d1$(x):b$=d2$(x):b=val(b$)
1560 return
1570 rem
1580 li=1:rem ligne
1590 co=1:rem col
1600 in=0:rem nbre d'exam
1610 gosub2360
1620 goto1810
1630 ifz$="▒"thenco=co+1:goto1810
1640 ifz$=chr$(13)orz$="▒"thenli=li+1:goto1810
1650 ifz$="▒"thenco=co-1:goto1810
1660 ifz$="▒"orz$=chr$(141)thenli=li-1:goto1810
1670 ifz$="▒"thenreturn
1680 ifz$="5"thenli=xl/2:co=xc/2:goto1810
1690 ifz$="▒"thengosub2360:in=0:li=1:co=1:goto1810
```

```
1700 ifz$="7"orz$="■"thenli=1:co=1:goto1810
1710 ifz$="3"thenli=xl:co=xc:goto1810
1720 ifz$="2"thenli=xl:co=xc/2:goto1810
1730 ifz$="1"thenli=xl:co=1:goto1810
1740 ifz$="9"thenli=1:co=xc:goto1810
1750 ifz$="4"thenli=xl/2:co=1:goto1810
1760 ifz$="6"thenli=xl/2:co=xc:goto1810
1770 ifz$="8"thenli=1:co=xc/2:goto1810
1780 ifz$=" "orz$="*"thenifpeek(p)<128thena$="":gosub1910:li=li+1:goto1810
1790 ifz$="-"thenifpeek(p)>127thena$="":gosub1970:li=li+1:goto1810
1800 ifz$="■"thenreturn
1810 li=int(li):co=int(co)
1820 ifli>xlthenli=1:co=co+1
1830 ifco>xcthenco=1:li=li+1:ifli>xlthenli=1
1840 ifli<1thenli=xl:co=co-1
1850 ifco<1thenli=li-1:co=xc:ifli<1thenli=1
1860 p=32768+40*(sl+li-1)+(co-1)*tz+ma
1870 a=peek(p)
1880 getz$:ifz$>""thenpokep,a:goto1630
1890 ifcp<10thencp=cp+1:goto1880
1900 cp=0:pokep,peek(p)+128+256*(peek(p)>127):goto1880
1910 ifin=tt+1thenreturn
1920 fori=ptop+vz-1:pokei,peek(i)+128+256*(peek(i)>127):next
1930 li%=li:co%=co:gosub1540:ifsttthenstop
1940 printleft$(li$+bl$,40)"■";
1950 li%(in)=li:co%(in)=co:in=in+1
1960 return
1970 fori=ptop+vz-1:pokei,peek(i)+128+256*(peek(i)>127):next
1980 li%=li:co%=co:gosub1540:ifsttthenstop
1990 li$="■"+li$+"■"
2000 printleft$(li$+bl$,42)"■";
2010 fori=0toin-1:ifli<>li%(i)orco<>co%(i)thennext:stop
2020 j=i:i=in:next
2030 fori=jtoin-2:li%(i)=li%(i+1):co%(i)=co%(i+1):next:in=in-1
2040 return
2050 print"■■■■■■■■■■■CODE des SERVICES"
2060 print"■■■■■■■■■■■
```

```
2300 return
2310 getq$:ifq$=""then2310:gosub2330
2320 return
2330 aq=asc(q$):ifaq>64andaq<91thenaq=aq-64:goto2350
2340 ifaq>48oraq<58thenaq=aq-22
2350 return
2360 rem
2370 poke59458,62
2380 print"▒   examens de sang       ] ur par bac     "
2390 print"────┬────┬────┬────┬────[────┬────┬────"
2400 print"Uree]NFVS]Magn]T3T4]Asly]Rech]CBUi]CBUs"
2410 print"Glyc]Hypr]HDL.]AnAu]W.Fe]Amyu]Copi]Cops"
2420 print"ChoT]Plaq]Ca.S]Lipg]Wrig]Elec]Myci]Mycs"
2430 print"SMA.]Reti]PhoS]Lith]BRhu]SPig]B.K.]Coba"
2440 print"Iono]Ptrh]Fer.]GazS]▄▄▄▄]Sang]Geni]Gens"
2450 print"AUri]Howl]Sidr]Myel]▄▄▄▄]Ph.U]Expi]Exps"
2460 print"LipT]ATh3]▄▄▄▄]Gtte]Cbsp]Ur.U]Lari]LArs"
2470 print"Trig]Refx]APyr]Thrb]▄▄▄▄]Crea]Pusi]Puss"
2480 print"Crea]TsTc]GrRh]GHLA]ClCr]Uriq]DSTi]DSTs"
2490 print"Bili]Cmbs]Phty]Lyse]Cate]PhoU]Gogi]Gogs"
2500 print"BLip]Aslo]PTtX]CoFc]Vany]Ca.U]Frot]Prol"
2510 print"GmGt]Fibr]NH3.]PDFc]Harg]Iono]Hemi]Hems"
2520 print"Palc]B.W.]BWqt]Nels]Le.T]Porp]Para]▄▄▄▄"
2530 print"GoGp]Rub1]Hang]CRP.]Cprt]▄▄▄▄]Scot]Dige"
2540 print"BHep]Tox1]MacL]Latx]TTrh]OHPr]Addi]Palu"
2550 print"LdhC]Aggl]Waal]Tox2]FSH.]Pgol]Cyto]BKex"
2560 print"Aldo]Dbg.]▄▄▄▄]Rub2]AAnu]Oest]17Co]Abio"
2570 print"PAci]Elec]Cu.S]ImEl]▄▄▄▄]L.H.]▄▄▄▄]Imug"
2580 print"Amyl]IgGM]Cu.U]Foet]LCRi]LCRs]LCRk]LCRe"
2590 print"────────────────────────────────────────────"
2600 return
2610 d1$(0)="uree":d2$(0)="10"
2620 d1$(1)="num.formule-v.s.":d2$(1)="38"
2630 d1$(2)="mg ser. et glob.":d2$(2)="40"
2640 d1$(3)="t3-t4 ":d2$(3)="xxx"
2650 d1$(4)="antistaphylolysines":d2$(4)="20"
2660 d1$(5)="alb-glu-acet-sang-ph":d2$(5)="20"
2670 d1$(6)="c.b.u.inf":d2$(6)="155"
2680 d1$(7)="c.b.u.st.":d2$(7)="45"
2690 d1$(8)="glycemie":d2$(8)="10"
2700 d1$(9)="hyperglycemie":d2$(9)="60"
2710 d1$(10)="hdl cholesterol ":d2$(10)="80"
2715 d1$(11)="antigene australia":d2$(11)="60"
2720 d1$(12)="widal felix":d2$(12)="40"
2730 d1$(13)="amylasurie":d2$(13)="30"
2740 d1$(14)="coprocult.inf.":d2$(14)="150"
2750 d1$(15)="coprocult.st.":d2$(15)="110"
2760 d1$(16)="cholesterol total":d2$(16)="10"
2770 d1$(17)="plaquettes":d2$(17)="15"
2780 d1$(18)="calcium sanguin":d2$(18)="20"
2790 d1$(19)="lipidogramme":d2$(19)="60"
2800 d1$(20)="wright":d2$(20)="40"
2810 d1$(21)="electro./prot./urine":d2$(21)="70"
2820 d1$(22)="mycose positive":d2$(22)="60"
2830 d1$(23)="mycose sterile":d2$(23)="35"
2840 d1$(24)="s.m.a":d2$(24)="125"
2850 d1$(25)="reticulocytes":d2$(25)="10"
2860 d1$(26)="phosphore sanguin":d2$(26)="15"
2870 d1$(27)="lithium":d2$(27)="20"
```

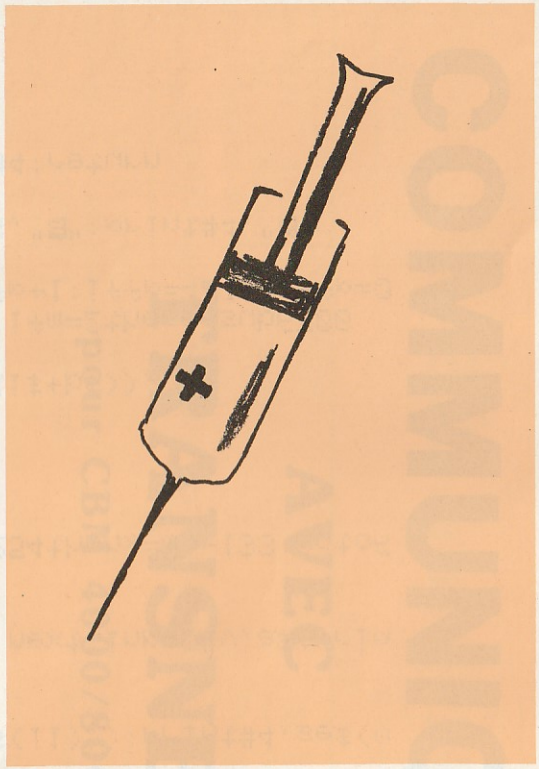
```
2880 d1$(28)="bilan du rhumatisme":d2$(28)="180"
2890 d1$(29)="sels-pig.bil.-uro.":d2$(29)="10"
2900 d1$(30)="xxxxx":d2$(30)="00"
2910 d1$(31)="cobaye":d2$(31)="90"
2920 d1$(32)="ionogramme +r.a.":d2$(32)="60"
2930 d1$(33)="prothrombine tck":d2$(33)="40"
2940 d1$(34)="fer serique+capacite":d2$(34)="50"
2950 d1$(35)="gaz du sang":d2$(35)="50"
2960 d1$(36)="xxxxx":d2$(36)="000"
2970 d1$(37)="sang":d2$(37)="10"
2980 d1$(38)="genitaux-positifs":d2$(38)="220"
2990 d1$(39)="genitaux st":d2$(39)="55"
3000 d1$(40)="acide urique":d2$(40)="10"
3010 d1$(41)="temps de howell-tck":d2$(41)="35"
3020 d1$(42)="siderophiline":d2$(42)="50"
3030 d1$(43)="myelogramme":d2$(43)="50"
3040 d1$(44)="xxxxx":d2$(44)="00"
3050 d1$(45)="phosphore urinaire":d2$(45)="15"
3060 d1$(46)="expecto. pos.b.k.":d2$(46)="260"
3070 d1$(47)="expecto.-b.k.st":d2$(47)="165"
3080 d1$(48)="lipides totaux":d2$(48)="20"
3090 d1$(49)="antithrombine iii ":d2$(49)="40"
3100 d1$(50)="xxxxxx":d2$(50)="00"
3110 d1$(51)="goutte epaisse":d2$(51)="25"
3120 d1$(52)="clearance b.s.p":d2$(52)="45"
3130 d1$(53)="uree urinaire":d2$(53)="10"
3140 d1$(54)="liq. articul. non ster.":d2$(54)="240"
3150 d1$(55)="liquide articulaire-st":d2$(55)="55"
3160 d1$(56)="triglycerides":d2$(56)="30"
3170 d1$(57)="reflexogramme":d2$(57)="50"
3180 d1$(58)="acide pyruvique":d2$(58)="50"
3190 d1$(59)="thromboelastogramme":d2$(59)="50"
3200 d1$(60)="xxxxxx":d2$(60)="00"
3210 d1$(61)="creatinine urine":d2$(61)="10"
3220 d1$(62)="pus non sterile ":d2$(62)="220"
3230 d1$(63)="pus-oeil-nez-st ":d2$(63)="55"
3240 d1$(64)="creatinine sang":d2$(64)="10"
3250 d1$(65)="temps de saign./coag.":d2$(65)="15"
3260 d1$(66)="groupe sang-rh-agglut":d2$(66)="80"
3270 d1$(67)="groupage h.l.a":d2$(67)="180"
3280 d1$(68)="clearance creatinine":d2$(68)="30"
3290 d1$(69)="acide urique urine":d2$(69)="10"
3300 d1$(70)="drain-sonde-tub.non st.":d2$(70)="220"
3310 d1$(71)="drain-sonde-tub.st":d2$(71)="55"
3320 d1$(72)="bilirubines":d2$(72)="25"
3330 d1$(73)="test de coombs":d2$(73)="30"
3340 d1$(74)="phenotype-rh":d2$(74)="20"
3350 d1$(75)="lyse des euglobulines":d2$(75)="25"
3360 d1$(76)="catecholamines":d2$(76)="60"
3370 d1$(77)="phosphore urinaire":d2$(77)="15"
3380 d1$(78)="cytobac gorge pos.":d2$(78)="200"
3390 d1$(79)="cytobac.gorge-st":d2$(79)="80"
3400 d1$(80)="bilan lipidique":d2$(80)="180"
3410 d1$(81)="antistreptolysines":d2$(81)="35"
3420 d1$(82)="protides totaux":d2$(82)="10"
3430 d1$(83)="cofacteurs-ii-v-vii+x":d2$(83)="50"
3440 d1$(84)="ac.vanylmand.":d2$(84)="60"
3450 d1$(85)="calcium urinaire":d2$(85)="20"
3460 d1$(86)="frottis":d2$(86)="70"
```

```
3470 d1$(87)="prolans b(r.i.g)":d2$(87)="100"
3480 d1$(88)="gamma-gt":d2$(88)="25"
3490 d1$(89)="fibrine":d2$(89)="20"
3500 d1$(90)="ammoniac":d2$(90)="50"
3510 d1$(91)="p.d.f. compl. sol.":d2$(91)="50"
3520 d1$(92)="cellules de hargrave":d2$(92)="20"
3530 d1$(93)="ionogramme urinaire":d2$(93)="35"
3540 d1$(94)="hemoculture infes.":d2$(94)="110"
3550 d1$(95)="hemoculture st.":d2$(95)="40"
3560 d1$(96)="phosphatases alcalines":d2$(96)="25"
3570 d1$(97)="b.w.":d2$(97)="20"
3580 d1$(98)="b.w.quanti":d2$(98)="40"
3590 d1$(99)="nelson":d2$(99)="85"
3600 d1$(100)="le test":d2$(100)="20"
3610 d1$(101)="porphyrines":d2$(101)="70"
3620 d1$(102)="parasitologie":d2$(102)="70"
3630 d1$(103)="xxxxx":d2$(103)="00"
3640 d1$(104)="transaminases g.o.-g.p.t.":d2$(104)="45"
3650 d1$(105)="rubeole":d2$(105)="30"
3660 d1$(106)="hanger":d2$(106)="10"
3670 d1$(107)="c.r.p.":d2$(107)="40"
3680 d1$(108)="consom./proth.":d2$(108)="30"
3690 d1$(109)="xxxxxx":d2$(109)="00"
3700 d1$(110)="scotch test":d2$(110)="10"
3710 d1$(111)="digestion-chimie":d2$(111)="70"
3720 d1$(112)="bilan hepatique":d2$(112)="210"
3730 d1$(113)="toxoplasmose":d2$(113)="40"
3740 d1$(114)="mac lagan":d2$(114)="10"
3750 d1$(115)="test  au  latex":d2$(115)="20"
3760 d1$(116)="temps de thrombine-reptilase":d2$(116)="30"
3770 d1$(117)="hydroxyproline":d2$(117)="70"
3780 d1$(118)="compte d'addis":d2$(118)="25"
3790 d1$(119)="rech.palu-para.":d2$(119)="15"
3800 d1$(120)="l.d.h.-c.p.k.":d2$(120)="60"
3810 d1$(121)="agglut.irregul.":d2$(121)="30"
3820 d1$(122)="waaler rose":d2$(122)="35"
3830 d1$(123)="toxoplasmose 2":d2$(123)="60"
3840 d1$(124)="f.s.h.":d2$(124)="100"
3850 d1$(125)="pregnandiol":d2$(125)="60"
3860 d1$(126)="cytologie":d2$(126)="15"
3870 d1$(127)="b.k. expecto.":d2$(127)="65"
3880 d1$(128)="aldolase":d2$(128)="30"
3890 d1$(129)="diagn.biol./grossesse":d2$(129)="30"
3900 d1$(130)="xxxxxx":d2$(130)="00"
3910 d1$(131)="rubeole deux. deter.":d2$(131)="45"
3920 d1$(132)="anticorps antinucl.":d2$(132)="30"
3930 d1$(133)="oestradiol":d2$(133)="80"
3940 d1$(134)="17oeto-17OH":d2$(134)="130"
3950 d1$(135)="antibiogrmme":d2$(135)="40"
3960 d1$(136)="phosphatases acides":d2$(136)="25"
3970 d1$(137)="electrophorese":d2$(137)="60"
3980 d1$(138)="ou sang ceruleoplasmine":d2$(138)="115"
3990 d1$(139)="immunoelectrophorese":d2$(139)="180"
4000 d1$(140)="electro.hemoglob.":d2$(140)="80"
4010 d1$(141)="l.h":d2$(141)="100"
4020 d1$(142)="xxxx":d2$(142)="00"
4030 d1$(143)="lor-immunoglob.(a.g.m.)":d2$(143)="200"
4040 d1$(144)="amylasemie":d2$(144)="25"
4050 d1$(145)="iga-igg-igm":d2$(145)="80"
```

```
4060 d1$(146)="cuivre urinaire":d2$(146)="50"
4070 d1$(147)="alphafoeto proteine":d2$(147)="40"
4080 d1$(148)="lcr(cytobac-chim)inf.":d2$(148)="155"
4090 d1$(149)="lcr(cytobact.chimie)st.":d2$(149)="55"
4100 d1$(150)="lcr(bk)":d2$(150)="50"
4110 d1$(151)="lcr(electro.+gamma-glob.)":d2$(151)="200"
4120 return
4130 print"■■■PRELEVEMENT ■o■ui/■n■on"
4140 getz$:ifz$="n"thenreturn
4150 ifz$<>"o"then4140
4160 gosub4230
4170 return
4180 print"■■■■■■■■■■■■■■■■examens speciaux  ■o■ui/■n■on"
4190 getz$:ifz$="n"thenreturn
4200 ifz$<>"o"then4190
4210 gosub4850
4220 return
4230 c1=0
4240 print"■■■■■■■■■■■■BAISIE DES PRELEVEMENTS":c1=c1+1
4250 print"■■Sang veineux/pli du coude.■1■ "
4260 print"■■sang multiples(au moins 4):■2■ "
4270 print"■■catheterisme uretral/femme:■3■ "
4280 print"■■tubage gastrique..........:■4■ "
4290 print"■■aseptique niveau muqeuse..:■5■ "
4300 print"■■gyneco.differents niveaux.:■6■ "
4310 kf(1)=1.5:kf(2)=4:kf(3)=1.25:kf(4)=2.25:kf(5)=1:kf(6)=3:vk=8.70:va=8.30
4320 print"■■■TAPEZ LE CODE■"
4330 getko:ifko<1orko>6then4330
4340 ko$(c1)=str$(ko)
4350 print"■■■■TAPEZ■":print"■■k■/medecin"
4360 print"■■kb■/biologiste"
4370 print"■■a■/infirmier (1,2,3,4)"
4380 print"■■b■/technicien capacitaire (1,2)"
4390 print" ■                                  "
4400 getz$:ifz$=""then4400
4410 ifz$="k"thenpx(c1)=kf(ko)*vk:goto4460
4420 ifz$="a"andko<5thenpx(c1)=kf(ko)*va:goto4460
4430 ifz$="a"thenprint"■NON AUTORISE■■■":goto4390
4440 ifz$="b"andko<3thenpx(c1)=kf(ko)*va:goto4460
4450 ifz$="b"thenprint"■NON AUTORISE■■■":goto4390
4460 px(c1)=int(100*px(c1))/100
4470 print:print"■encore un prelevement ■■■o■ui/■n■on"
4480 getz$
4490 ifz$="o"then4240
4500 ifz$<>"n"then4480
4510 px=0:forj=1toc1:px=px+px(c1):px$=mid$(str$(px),2):next:print"Le prix est
px$
4520 return
4530 open4,4,1:print#4,d1$
4540 print#4
4550 print#4,"    "chr$(1)mj$no$
4560 print#4,"    "chr$(1)mj$left$(pr$+bl$,17)sx$"  "dn$
4570 print#4,mj$"No d'Hospit:"ho$
4580 print#4,mj$"Service : "sv$(aa)
4590 print#4
4600 forj=0toin-1:li%=li%(j):co%=co%(j):gosub1540:print#4,li$b" ";:next
4610 print#4
4620 print#4,mj$"Total : "t;" B"
```

```
4630 print#4
4640 ifc1>0thenfori=1toc1:print#4,ko$(i)"  ";:next:print#4,"total prelevement=
px$
4650 print#4
4660 ifc2>0thenfori=1toc2:print#4,sa$(i)" ";:next
4670 ifc3>0thenfori=1toc3:print#4,ns$(i)" "ix(i);:next
4680 print#4:close4
4690 return
4700 open4,4,1
4705 fori=1toea:print#4:print#4,mj$left$(no$+"  "+pr$+" "+sx$+bl$,30)dj$right$
"000"+n0$,3)
4710 print#4
4720 print#4:print#4,ru$
4730 print#4,left$(cp$+" "+vi$+bl$,25)"dr: "md$
4740 next:close4
4750 return
4760 fg$=fs$+dj$:open2,8,2,e0$++fg$+ws$
4770 print#2,"#"c$;"0"c$;
4780 close2:return
4790 open2,8,2,e0$++fg$+ws$:open3,8,3,d0$++fg$+rs$
4800 input#3,a$:if left$(a$,1)=mc$then4800
4810 ifa$<>"#"thenprint#2,a$c$;:goto4800
4820 print#2,cc$c$;
4830 print#2,"#"c$;mid$(str$(no),2)c$;
4840 close2:close3:return
4850 c2=0
4860 print"■■■■■■■■■■■■SAISIE DES EXAMENS SPECIAUX"
4870 print"■■Phenobarbital                : ■01■ "
4880 print"Phenytoine(dihydan)......: ■02■ "
4890 print"Tegretol..................: ■03■ "
4900 print"Zarontin,ethynal..........: ■04■ "
4910 print"Mysoline..................: ■05■ "
4920 print"Morphine/urines...........: ■06■ "
4930 print"Cocaine/urines............: ■07■ "
4940 print"Amphetamines/urines......: ■08■ "
4950 print"Barbituriques/urines.....: ■09■ "
4960 print"Barbituriques/urines.....: ■10■ "
4970 print"Methadone..../urines.....: ■11■ "
4980 print"..........................: ■12■ "
4990 print"..........................: ■13■ "
5000 print"..........................: ■14■ "
5010 print"..........................: ■15■ "
5020 print"..........................: ■16■ "
5030 print"■■TAPEZ LE CODE■ ■■■'exa.noncode"
5040 inputko$:ko=val(ko$)
5050 ifko$="@"thengosub5190:goto5110
5060 ifko<1orko>20thenprint"■                                    ":goto5040
5070 c2=c2+1:sa$(c2)=ko$
5080 print"■■■On vient de saisir l'examenspecial":print
5090 printtab(5)se$(val(sa$(c2)))
5100 printtab(5)left$(br$,len(se$(val(sa$(c2)))))
5110 print"■■ENCORE UN EXAMEN SPECIAL ■■■■■■■■■■■■
```

```
5180 return
5190 c3=0
5200 c3=c3+1:print"ENTREZ LE NOM":inputns$(c3)
5210 print"entrez le Prix":inputix(c3):ix=ix+ix(c3)
5220 print"encore oui/non""
5230 getz$:ifz$="o"then5200
5240 ifz$<>"n"then5230
5250 ix$=mid$(str$(ix),2):return
5260 open2,8,2,fg$+rs$
5270 input#2,a$:ifa$<>"#"thenprinta$:goto5270
5275 input#2,a$:printa$
5280 close2:return
5290 fori=1to28:printl$(i)" ";:next:return
5360 open2,8,2,d0$++fg$+rs$
5370 input#2,a$:ifa$<>"#"thengosub5800:goto5370
5376 input#2,a$:no=val(a$)
5380 close2:return
5500 print"
5510 input"quelle date xx.09.81"da$
5520 fg$="fs"+da$:gosub5360:goto600
5600 print"
5610 input"quelle date xx.09.81"da$
5620 fg$=d0$+"fs"+da$:gosub5260
5630 print"TAPEZ SHIFT pour Continuer"
5640 wait152,1:print":goto50
5700 remprint"entrez votre code"
5710 reminputcd$
5720 print:input"la date xx.08.81"da$
5725 fg$=fs$+da$
5730 gosub5360
5740 fori=1to29:ifl$(i)<>""thenk=k+1:aa(k)=i:print"no"k;" "av$(i),
5745 next:input"no debut";d1
5746 input"no fin";d2
5748 open4,4:print#4,"
5750 fori=d1tod2:ar=aa(i):n1%=len(l$(ar))/4:fj$=left$(l$(ar),1)
5755 forl=0ton1%-1:n0$=mid$(l$(ar),3+4*l,3):fi$=fj$+mid$(l$(ar),2+4*l,1)
5760 gosub6120:next
5762 iffp<>0thenprint#4,"":print#4,":fp=0
5764 next
5770 goto60
5800 a$=left$(a$,1):gosub2330:l$(aa)=a$:return
6000 print"ENTREZ votre CODE"
6005 getz$:ifz$=""then6005
```



```
6010 ifz$=" "then60
6020 ifz$<>left$(cd$,1)then60
6030 getz$:ifz$=""then6030
6040 ifz$<>mid$(cd$,2,1)then60
6050 print"[illegible]TERMINE[illegible]":end
6120 openv,8,v,d0$+fi$+n0$+rs$:t=0:m=1:lx$(1)="":lx$(2)=""
6130 input#v,a0$,a1$,a2$,a3$,a4$,a5$,a6$,a7$,a8$,a9$,b1$,b2$,b3$
6150 print#4,chr$(1)+a2$+chr$(129)"    "a3$"    "a6$"    "a1$"    "a4$"    "chr$(17
)sv$(ar)
6155 print#4,a7$,chr$(17)"Tot.B"a9$,a0$
6160 c1=val(b1$):c2=val(b2$):c3=val(b3$)
6170 ifc1>0thenprint#4,"prelev.";:fori1=1toc1:input#v,a1$(i1):print#4,mj$+a1$(i
1)" ";:next
6172 ifc1>0theninput#v,a$:print#4,"prix="a$:s1=val(a$)
6180 s2=0:ifc2>0thenfori1=1toc2:input#v,a2$(i1):v2=(val(a2$(i1))):print#4,se$(v
2)" "sf$(v2)
6182 ifc2>0thens2=s2+val(sf$(v2)):next
6190 ifc3>0thenfori1=1toc3:input#v,a3$(i1):print#4,a3$(i1):next:input#v,a$:prin
t#4,"prix="a$
6200 input#v,a$:rem print#4,a$
6210 forj=1tolen(a6$):ww=asc(mid$(a6$,j,1)):ifww>175andww<254thenx=ww-133:gotoG
250
6220 ifww=254thenx=33:goto6250
6230 ifww=255thenx=12:goto6250
6240 x=ww-1
6250 gosub1550:print#4,left$(bl$,10)li$b$left$(bl$,30-len(li$+b$));
6251 t=t+b:nextj
6255 print#4:print#4,"prix total="t;"+"s2;"+"ix;"="t+s2+ix:ifm=2thengosub6500
6256 rem'frecordi1=1to5:pbackup4:next:lg=lg+8+int(j/2):fp=fp+1:iffp=4tleft$fp=0
gocopy6600:lg=0
6257 fori1=1to5:print#4:next:fp=fp+1:iffp=4thenfp=0:print#4,"[illegible]":print#4,"[illegible]"
6260 closev:return
6500 print#4:print#4:print#4:print#4:print#4:print#4:print#4:return
10000 open15,8,15:open2,8,2,"0:fs24,10,80,s,r"
10010 get#2,a$:printa$;:ifst=0then10010
15000 open15,8,15:open2,8,2,"0:fs24,10,80,s,r"
15010 get#2,a$:printa$;:ifst=0then15010
30000 input"lettres/fichier malade";fi$
30010 input"chiffres/fichier malade[illegible]001[illegible]";n0$
30020 gosub100:open4,4:gosub6120
30030 close4
```

COMMUNIQUEZ

AVEC

TRANSNEX

pour CBM 4000/8000

- Transnex asynchrone : carte d'interface RS232C/V24 connectée sur bus extension mémoire + extension Basic communications.

- Commodial : logiciel d'accès banques de données (QUESTEL...) services de messagerie électronique (MISSIVE...) livré avec programme source.

En option : adaptateur MINITEL permettant de relier un CBM équipé de TRANSNEX à un terminal MINITEL.

En option : adaptation CBM 3000 / 4000 petit écran.

DOMAINES D'UTILISATION:

- Accès banques de données sur réseau téléphonique ou TRANSPAC (jusqu'à 1200 bauds full duplex)
- Services de messages électroniques
- Transfert de fichiers entre CBM
- Transfert de fichiers de texte TRAITEXT vers télécomposeuse (logiciel COMPODIAL)
- Liaison à distance avec un autre ordinateur
- Liaisons locales
- Interfaçage de périphériques spéciaux V24 ou boucle de courant (multipoints) (crayon-lecteur, horloges pointeuses, balances de précision etc...)

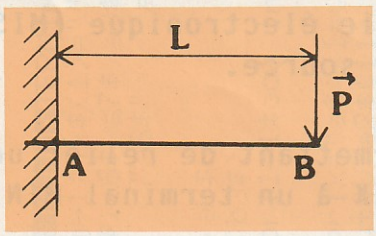
19-21, RUE MATHURIN-RÉGNIER - 75015 PARIS
TÉL. : 306.82.02 - TÉLEX : 204 875 F.

Flexion 2.5

Une application du CBM en calcul scientifique

I-Présentation

Ce programme calcule une poutre en flexion d'après le schéma suivant :



Il s'agit d'une poutre de longueur L, encastrée à une de ses extrémités (A), et à laquelle on applique une charge P à l'autre extrémité (B)

II - Caractéristiques du programme

Langage : Basic

Place mémoire requise : 8 K

Ecrit pour : CBM Série 3000

III - Organisation du programme

1) Lignes 0 à 16

Présentation et attente pour la suite.

Remarque : attente d'une touche par

POKE 158,0 : vide le buffer du clavier
WAIT 158,1 : attend l'appui d'une touche

2) Lignes 17 à 34

Conversation avec l'utilisateur

Entrée des données
1) charge appliquée
2) longueur de la poutre

Puis choix de la section de la poutre.

3) Lignes 35 à 39

Branchements suivant la section choisie.
Si il y a un mauvais code, on prend par défaut la section elliptique.

4) Lignes 40 à 46

Section elliptique
Entrée des données petit axe
grand axe
Calcul du moment d'inertie
Branchement à la ligne 70

5) Lignes 47 à 53

Section Rectangulaire

Données Hauteur
Base
Calcul Moment d'inertie
Branchement en 70

6) Lignes 54 à 58

Section circulaire pleine
Données Diamètre
Calcul Moment d'inertie
Branchement en 70.

7) Lignes 59 à 65

Section circulaire creuse
Données Diamètre extérieur
Diamètre intérieur
Calcul Moment d'inertie
Branchement en 70

8) Lignes 66 à 69

Section hexagonale
Données Largeur
Calcul Moment d'inertie

9) Lignes 70 à 73

Calcul du moment de flexion maxi.
Calcul de la contrainte admissible.
Appel du sous-programme Mater. (Matériaux).

10) Ligne 74

Test pour savoir si le problème posé respecte les conditions de résistance.

11) Lignes 75 à 91

Calcul de la flèche maxi.
Affichage des résultats.

12) Lignes 92 à 108

On peut changer des données.

13) Lignes 109 à 116

Calcul de la Force.
Données Flèche maxi imposée
longueur de la poutre
Calcul
Branchement en 76 pour affichage.

14) Lignes 117 à 123

Calcul de la longueur
Donnée Nouvelle charge
Valeur de la flèche
Calcul
Branchement en 76 pour affichage

15) Lignes 124 à 132

Changement de section
Données Nouvelle section

16) Lignes 133 à 148

Section rectangulaire
Données Domaine de variation de la base
Calcul
Branchement en 76 pour affichage.

17) Lignes 149 à 164

Section circulaire creuse
Données Diamètre ext.Mini
Diamètre ext.Maxi
Calcul
Branchement en 76 pour affichage.

18) Ligne 165 à 179

Section elliptique
Données Valeur inf. du grand axe
Valeur sup. du grand axe
Calcul
Branchement en 76 pour affichage.

19) Lignes 180 à 199

SBR Mater
Données choix du Matériau

20) Lignes 200 à 204

Problème ne satisfaisant pas aux conditions de résistance.

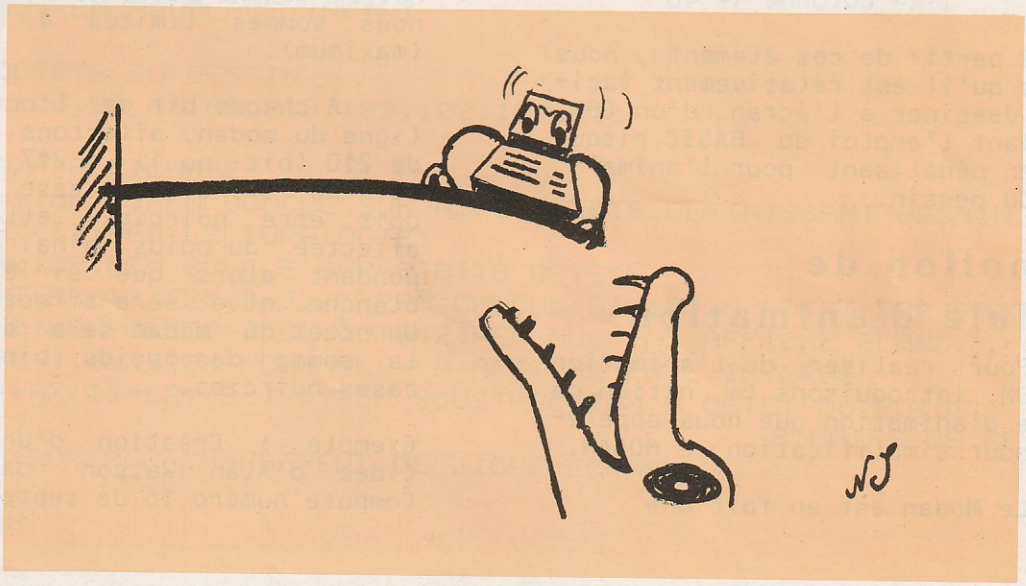
J.BRUGEASSOU

```
0 POKE59490,62:P0=5*SQR(3)
1 PRINT"▯▯▯──────────────────────────────────────────────────────────────"
2 PRINT"▯▯   PROGRAMME DE CALCUL DE FLEXION
3 PRINT"                                                                  "
4 PRINT"          ▒   L
5 PRINT"          ▒<──────────>|
6 PRINT"          ▒            | P
7 PRINT"          ▒            ▼
8 PRINT"          ▒ ▬▬▬
```

```
57 V=P1/2
58 GOTO70
59 INPUT" VALEUR DU DIAMETRE EXT     0■■■■";P1
60 IFP1=0THENPRINT"■■":GOTO59
61 INPUT" VALEUR DU DIAMETRE INT     0■■■■";P2
62 IFP2=0THENPRINT"■■":GOTO61
63 I=PI*(P1*P1*P1*P1-P2*P2*P2*P2)/64
64 V=P1/2
65 GOTO70
66 INPUT" VALEUR DE LA LARGEUR     0■■■■";P1:P2=0
67 IFP1=0THENPRINT"■■":GOTO66
68 I=P0*P1*P1*P1*P1/144
69 V=P1/2
70 :
71 MF=P*L
72 SI=MF/(I/V)
73 GOSUB180:REM SBR  MATER
74 IFSI>=RPTHENGOTO200
75 FM=P*L*L*L/3/E/I
76 PRINT"■■■■
```

```
117 INPUT"▯▯▯ NOUVELLE CHARGE    0▮▮▮▮";Z0
118 IFZ0=0THENPRINT"▯▯":GOTO117
119 INPUT"▯ VALEUR DE LA FLECHE    0▮▮▮▮";Z1
120 IFZ1=0THENPRINT"▯▯":GOTO119
121 Z2=Z1*3*E*I/Z0
122 L=Z2↑(1/3)
123 FM=Z1:P=Z0:GOTO76:                    REM AFFICHAGE
124 PRINT"▯▯▯  VOTRE SECTION EST "P$
125 PRINT"▯▯▯ CODE ▯RE▮▮ ▯CI▮▮ ▯CC▮▮ ▯EX▮▮ ▯EL▮▮ ";
126 INPUT"    X▮▮▮▮";A$
127 IFA$="X"THENPRINT"▯▯":GOTO125
128 IFA$="RE"THEN133
129 IFA$="CI"THEN149
130 IFA$="CC"THEN152
131 IFA$="EX"THEN165
132 IFA$="EL"THEN167
133 PRINT" DOMAINE DE VARIATION DE LA BASE"
134 INPUT"  VALEUR INF     0▮▮▮▮";INF
135 IFINF=0THENPRINT"▯▯":GOTO134
136 INPUT"  VALEUR SUP     0▮▮▮▮";SUP
137 IFSUP=0THENPRINT"▯▯":GOTO136
138 PRINT"▯▯▯ RESULTATS
139 PRINT" ──────────"
140 PRINT"BASE",,"HAUTEUR"
141 PRINT"────",,"────────"
142 FORI=INFTOSUPSTEP.5
143 H3=P*L*L*L*4/(E*FM*I):H=H3↑(1/3)
144 PRINTI,,FNF(H)
145 NEXT
146 P1=SUP:P2=H
147 P$="RECTANGULAIRE":P1$="BASE":P2$="HAUTEUR"
148 PRINT"UNE TOUCHE SVP":WAIT158,1:POKE158,0:GOTO76
149 P$="CIRCULAIRE PLEIN":P1$="DIAMETRE EXT":P2$="":P2=0
150 Z4=P*L*L*L*64/(3*E*FM*PI):Z=Z4↑0.25
151 P1=Z:GOTO76
152 PRINT" VARIATION DU DIAMETRE"
153 INPUT"          EXT MINI    0▮▮▮▮";INF
154 IFINF=0THENPRINT"▯▯":GOTO153
155 INPUT"          EXT MAXI    0▮▮▮▮";SUP
156 IFSUP=0THENPRINT"▯▯":GOTO155
157 PRINT"DIAMETRE EXT","DIAMETRE INT"
158 PRINT"──────────","──────────"
159 FORI0=INFTOSUP
160 P$="CIRCULAIRE CREUSE":P1$="DIAMTRE EXT":P2$="DIAMETRE INT"
161 Z4=I0*I0*I0*I0-((P*L*L*L*64)/(3*E*FM*PI)):Z=ABS(Z4)↑.25:P1=I0:P2=Z
162 PRINTP1,P2
163 NEXT
164 PRINT"UNE TOUCHE SVP":WAIT158,1:POKE158,0:GOTO76
165 Z4=P*L*L*L*144/(15*E*FM*SQR(3)):P1=Z4↑.25:P$="EXAGONNALE":P1$=
166 P2$="":P2=0:GOTO76                                    "LARGEUR"
167 PRINT"DOMAINE DE VARIATION DE GRAND AXE"
168 P$="ELIPTIQUE":P1$="GRAND AXE":P2$="PETIT AXE"
169 INPUT"                VALEUR INF.   0▮▮▮▮";INF
170 IFINF=0THENPRINT"▯▯":GOTO169
171 INPUT"                VALEUR SUP.   0▮▮▮▮";SUP
172 IFSUP=0THENPRINT"▯▯":GOTO171
173 PRINT"GRAND AXE",,"PETIT AXE"
174 PRINT"─────────",,"──────────"
175 FORI0=INFTOSUP
176 Z3=P*L*L*L*4/(3*E*FM*PI*I0):P1=I0:P2=Z3↑(1/3)
```

```
177 PRINTP1,,P2
178 NEXT
179 PRINT"UNE TOUCHE SVP":WAIT158,1:POKE158,0:GOTO76
180 PRINT"▯▯▯┌──────┬──────┬──────┬──────┐
181 PRINT"| CODE  | MATERIAU| YOUNG | RP   |
182 PRINT"├──────┼──────┼──────┼──────┤
183 PRINT"|   1   | ACIER   |210000 | 165  |
184 PRINT"├──────┼──────┼──────┼──────┤
185 PRINT"|   2   | FONTE   |100000 |  55  |
186 PRINT"├──────┼──────┼──────┼──────┤
187 PRINT"|   3   | CUIVRE  |110000 |  54  |
188 PRINT"├──────┼──────┼──────┼──────┤
189 PRINT"|   4   | BRONZE  | 90000 |  90  |
190 PRINT"└──────┴──────┴──────┴──────┘
191 PRINT"▯▯CODE DE VOTRE CHOIX"
192 GETRP$:IFRP$=""THEN192
193 A=VAL(RP$)
194 IFA<1ORA>4THEN192
195 ONAGOTO196,197,198,199
196 M$="ACIER":E=2.1E5:RP=165:RETURN
197 M$="FONTE":E=1E5:RP=55:RETURN
198 M$="CUIVRE":E=1.1E5:RP=54:RETURN
199 M$="BRONZE":E=.9E5:RP=90:RETURN
200 PRINT"▯▯▯  LE PROBLEME POSE NE SATISFAIT
201 PRINT"     PAS AUX CONDITIONS DE
202 PRINT"          .RESISTANCE.
203 PRINT"▯▯▯▯▯▯▯"
204 STOP
205 REM-----------------------------
206 REM      (C) BRUGEASSOU J.
207 REM      ================
208 REM        LE 28/10/1981
209 REM
210 REM        7000  OCTETS
211 REM         VERSION 2.5
212 REM-----------------------------
```



Dessin Animé sur CBM 3032 - Edex

Avec ces "modules d'animation" le CBM n'a rien à envier au 64 !

L'image-écran du CBM 3000

Une image-écran se définit par une matrice de 25 lignes de 40 colonnes soient 1000 "points". La mémoire est située dans un CBM 3000 entre 32768 et 33767.

A tout moment, on peut imposer le contenu d'une case-écran par un POKE x, y
avec:
x = adresse de la case
y = code-écran du symbole

Comment déterminer simplement l'adresse d'une case-écran ?

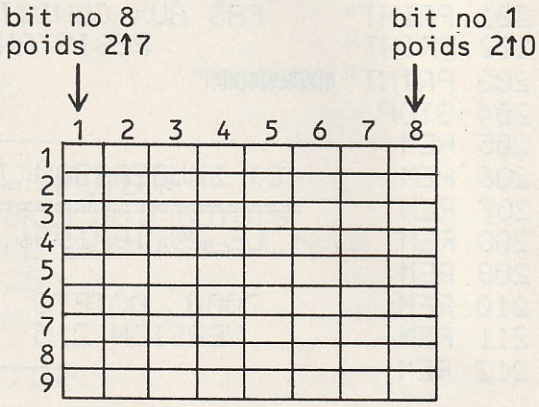
Adresse = 32768 +(40*(Ligne-1)) + (Colonne-1)
avec
1 <= ligne <= 25
1 <= colonne <= 40

A partir de ces éléments, nous voyons qu'il est relativement facile de dessiner à l'écran d'un CBM. Cependant l'emploi du BASIC risque d'être pénalisant pour l'animation du dessin.

La notion de module d'animation

Pour réaliser de l'animation sur CBM, introduisons la notion de module d'animation que nous appelerons pour simplification le MODAN.

Le Modan est en fait une grille de 9 lignes de 8 colonnes (matrice 9 x 8).



Pourquoi 8 colonnes? parce que cela revient à manipuler simplement un octet pour définir une ligne de notre modan. Volontairement, nous nous sommes limités à 9 lignes (maximum).

A chaque bit de l'octet d'une ligne du modan, affectons un poids de 2↑0 (bit no 1) à 2↑7 (pour le bit no 8). Si une case du modan doit être noircie, elle sera affectée du poids binaire correspondant alors que si elle reste blanche, elle sera affectée de 0. Un octet du modan sera défini par la somme des poids binaires des cases noircies.

Exemple : Création d'un bonhomme (idée d'Alan Watson parue dans Compute numéro 16 de septembre 81)

```
1 : 0  +64 +32 +16 +8 +4 +2 +0 =126
2 : 0  +64 +0  +16 +8 +0 +2 +0 = 90
3 : 0  +64 +0  +0  +0 +0 +2 +0 = 66
4 : 0  +0  +32 +16 +8 +4 +0 +0 = 60
5 :128 +64 +0  +16 +8 +0 +2 +1 =219
6 :128 +0  +32 +16 +8 +4 +0 +1 =189
7 : 0  +64 +32 +0  +0 +4 +2 +0 =102
8 :     "       "        "      "
9 :128 +64 +32 +0  +0 +4 +2 +1 =231
```

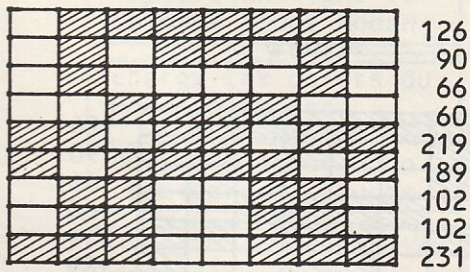
Nous voyons qu'un "bonhomme" peut être défini simplement à l'aide de 9 octets. Pour le positionner sur l'écran du CBM, il suffira de donner l'adresse de la première case du modan (case de la ligne 1 et de la colonne 1 du modan).

Nous allons afficher notre bonhomme à l'aide d'un programme BASIC. Ce programme est volontairement alourdi pour expliciter la façon de créer le dessin.

Nous obtenons bien notre bonhomme comme prévu à l'écran. Tentons maintenant à partir de ce programme BASIC une animation de notre bonhomme (exécution d'un sautillement d'une jambe à l'autre).

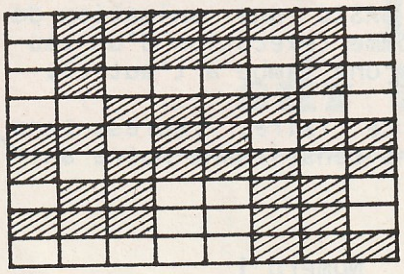
Pour ce faire, représentons les 3 modans constituant notre animation.

Numéro 1



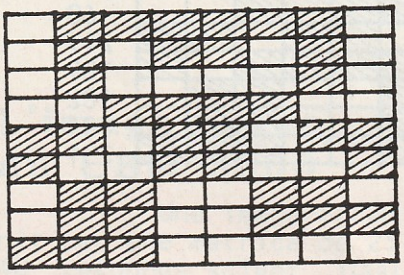

```
11 REM*********************************
12 REM* AFFICHAGE BASIC D'UN MODAN *
13 REM*  COPYRIGHT A.HESBOIS 07/82 *
14 REM*                             *
15 REM*      DESSIN DE BONHOMME     *
16 REM* ALAN WATSON REVUE COMPUTE   *
17 REM*      NUMERO 16 JUIN 1981    *
18 REM*                             *
19 REM*********************************
20 :
21 B%=32:IN%=160:QU%=40
22 PRINT"▯ ▨  EXEMPLE DE MODAN  ":PRINT" ▨ BONHOMME DE WATSON "
23 :
24 LI=7:CO=7:REM POSITION ECRAN LIGNE-COLONNE
25 ED=32768+(40*(LI-1))+(CO-1):REM CALCUL ADRESSE DEPART MODAN
26 :
27 NO%=9:REM NBRE OCTETS DU DESSIN
28 GOSUB 34
29 GOTO29:REM BOUCLAGE INFINI
30 :
31 REM OCTETS DU DESSIN
32 DATA 126,90,66,60,219,189,102,102,231
33 :
34 REM S/P AFFICHAGE
35 FORI=1TONO%:REM BOUCLE SUR NBRE OCTETS DEFINISSANT DESSIN
36 READA%(I):REM LECTURE OCTET
37 FORJ=0TO8:REM BOUCLE SUR 8 BITS OCTET
38 R%=INT(A%(I)/(2↑(8-J))):REM CALCUL DIVISION OCTET PAR POIDS
39 IFR%=0THENPOKEED+J,B%:GOTO42:REM NUL --> AFFICHE BLANC     BIT
40 POKEED+J,IN%:REM NON NUL --> AFFICHE PAVE INVERSE
41 A%(I)=A%(I)-(2↑(8-J)):REM SOUSTRACTION A OCTET DU POIDS
42 NEXT J                                          BINAIRE DE CE BIT
43 ED=ED+QU%:REM CHANGEMENT DE LIGNE ECRAN
44 NEXT I
45 RETURN
```

Numéro 2



126
90
66
60
219
189
102
230
7

Numéro 3



126
90
66
60
219
189
102
103
224

Nous voyons que pour animer le bonhomme, il suffit de 3 modans et dans ces 3 modans, seuls les 2 derniers octets varient :

Notre programme aura la structure suivante :

Modan numéro 1
↓
Modan numéro 2 ←
↓
Modan numéro 3

Les modans numéro 2 et numéro 3 n'auront que deux octets de définition puisque le reste de l'image du modan est inchangé.

Nous voyons que le programme BASIC ci-dessous conduit bien aux images prévues, mais l'animation n'a pas l'effet escompté, ce qui est dû à la lenteur de BASIC.

```
11 REM*********************************
12 REM* ANIMATION BASIC DU MODAN    *
13 REM*  COPYRIGHT A.HESBOIS 07/82 *
14 REM*                             *
15 REM*      DESSIN DE BONHOMME     *
16 REM* ALAN WATSON REVUE COMPUTE   *
17 REM*      NUMERO 16 JUIN 1981    *
18 REM*                             *
19 REM*********************************
20 :
21 B%=32:IN%=160:QU%=40
22 PRINT"❑ ■  ANIMATION DE MODAN  ":PRINT" ■  BONHOMME DE WATSON  "
23 :
24 LI=7:CO=7:ED=32768+(40*(LI-1))+(CO-1):NO%=9
25 READA$:IFA$<>"M1"THEN25:ELSEGOSUB39
26 RESTORE
27 LI=14:CO=7:ED=32768+(40*(LI-1))+(CO-1):NO%=2
28 READA$:IFA$<>"M2"THEN28:ELSEGOSUB39
29 RESTORE
30 LI=14:CO=7:ED=32768+(40*(LI-1))+(CO-1):NO%=2
31 READA$:IFA$<>"M3"THEN31:ELSEGOSUB39
32 GOTO26
33 :
34 REM OCTETS DES MODANS
35 DATA M1,126,90,66,60,219,189,102,102,231
36 DATA M2,230,7
37 DATA M3,103,224
38 :
39 REM S/P AFFICHAGE
40 FORI=1TONO%:READA%(I):FORJ=0TO8:R%=INT(A%(I)/(2↑(8-J)))
41 IFR%=0THENPOKEED+J,B%:GOTO42:ELSEPOKEED+J,IN%:A%(I)=A%(I)-(2↑(8-J))
42 NEXT:ED=ED+QU%:NEXT:RETURN
```

Aussi, allons nous écrire la partie d'affichage du modan en langage machine. Ce sous-programme sera appelé depuis BASIC par le biais de la fonction CALL disponible avec Edex 2.0.

La fonction CALL d'Edex 2.0

Nous allons expliciter quelque peu la fonction CALL (D'après la notice d'Edex 2.0 distribuée par Procep).

La fonction CALL permet l'appel d'un sous-programme écrit en langage machine commençant à une adresse A (exprimée en décimal) et ce, en passant des variables à la manière de la fonction USR.

La puissance supplémentaire apportée par CALL est que ces variables peuvent être réelles, entières ou des chaînes de caractères et qu'il est possible dans un même appel à un sous-programme de transférer jusqu'à 16 variables.

Syntaxe: CALL A, P1, P2,...P16

où les Pi désignent le nom des variables de type réel (A,Z...) de type entier (A%, Z%...) ou chaine caractères (A$, Z$...)

Au moment de l'exécution de CALL, les descriptions des variables nommées dans le CALL sont transférées dans le buffer du Basic du CBM à partir des adresses $0210. Ces descriptions sont faites par blocs de 4 octets. La première variable est décrite de $0210 à $0213 la seconde de $0214 à $0217,...

Nous allons travailler uniquement sur des variables entières. Dans ce cas les 4 octets définissent :

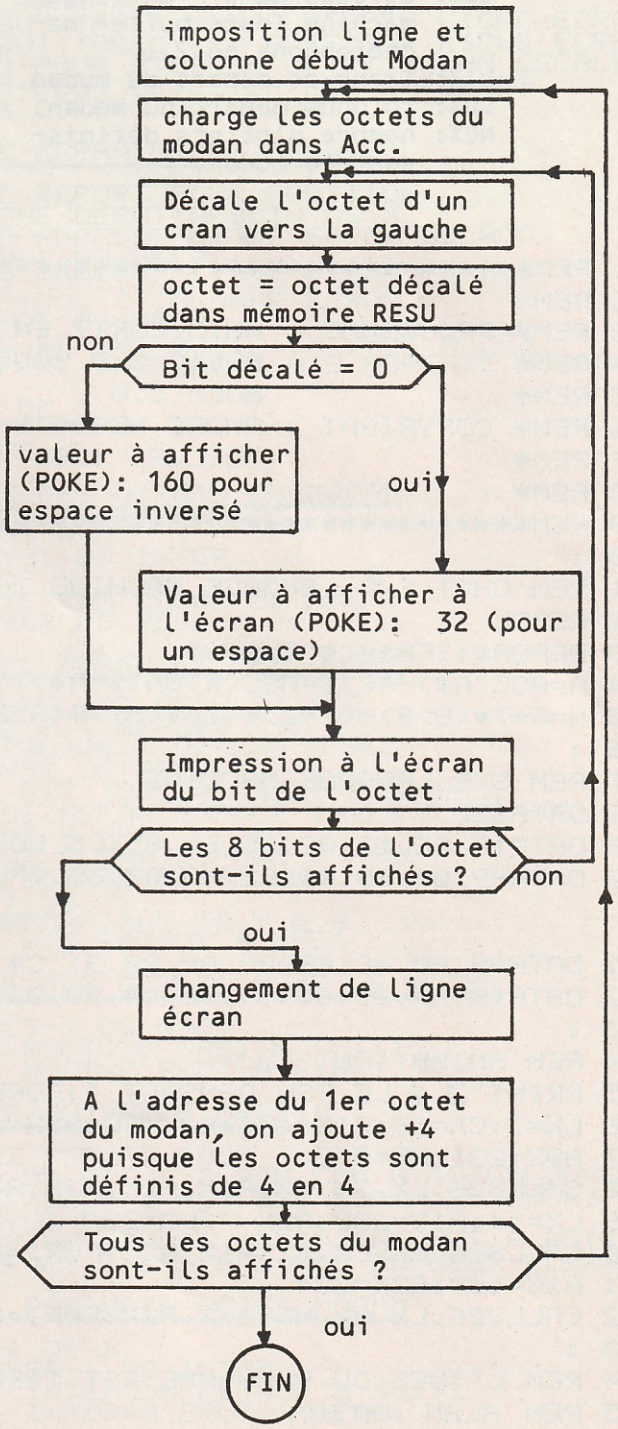
- le type de variable (1er octet) $80 pour variable entière

- indifférent pour le 2ème octet dans le cas d'une variable entière

- la valeur signée sur 2 octets (3ème et 4ème octet) de la variable entière.

Attention cette valeur est exprimée sous la forme poids fort-poids faible représentation classique des entiers dans le CBM.

Ordinogramme du sous-programme



Listing du sous-programme MODANIM

Avant de rentrer le programme ou de le lancer, faire SYS 46000 pour activer Edex 2.0.

Dans le programme BASIC, l'appel se fera comme suit:

CALL 826,Li%,CO%,NO%,X1%,X2%,..,X9%

826: adresse du S/P en langage machine (dans buffer magnétophone no 2).
LI%: ligne de départ du modan.
CO%: colonne départ du modan.
NO%: nombre d'octets définissant le modan.
Xi%: valeur du ième octet définissant le modan (nous aurons 9 valeurs maxi.).

Les variables passées par CALL seront définies aux adresses suivantes :

LI% ⟶ $0213
CO% ⟶ $0217
NO% ⟶ $021B
X1% ⟶ $021F
X2% ⟶ $0223
X3% ⟶ $0227
.
.
X9% ⟶ $023F

André HESBOIS

```
11 REM*********************************
12 REM*                               *
13 REM* PROGRAMME  : ANIM ECRIT EN    *
14 REM*              BASIC 3.0 SOUS   *
15 REM*              EDEX 2.0         *
16 REM* COPYRIGHT  : ANDRE HESBOIS    *
17 REM*              JUILLET 1982     *
18 REM*                               *
19 REM*********************************
20 :
21 REM CHGT S/P LANGAGE MACHINE
22 READL
23 READA$:IFA$="*"THEN35
24 A=ASC(A$)-48:B=ASC(RIGHT$(A$,1))-48:C=LEN(A$)
25 N=B+7*(B>9)-(C=2)*(16*(A+7*(A>9))):POKEL,N:L=L+1:GOTO23
26 :
27 REM S/P LANGAGE MACHINE
28 DATA826
29 DATA18,AE,13,02,B5,E0,85,C5,BD,48,E7,6D,17,02,85,C4,90,02,E6,C5,A2,00
30 DATAA9,04,8D,49,02,A0,00,BD,1F,02,8D,48,02,AD,48,02,18,0A,8D,48,02,90
                                                                    ,05
31 DATAA9,A0,4C,6E,03,A9,20,91,C4,C8,C0,08,D0,E8,CE,1B,02,D0,03,4C,91,03
32 DATA18,A9,28,65,C4,85,C4,90,02,E6,C5,E8,CE,49,02,D0,FA,4C,50,03,60,*
33 :
34 REM ANIMATION
35 PRINT"◘ ◙ LE FOU DANSANT ":FORB=1TO20
36 L%=7:C%=16:A1%=126:A2%=90:A3%=66:A4%=60:A5%=219:A6%=189:A7%=102:A8%=
37 A9%=231:N%=9                                                    A7%
38 CALL826,L%,C%,N%,A1%,A2%,A3%,A4%,A5%,A6%,A7%,A8%,A9%:FORI=1TO50:NEXT
39 L%=14:A1%=230:A2%=7:N%=2
40 CALL826,L%,C%,N%,A1%,A2%:FORI=1TO50:NEXT
41 A1%=103:A2%=224
42 CALL826,L%,C%,N%,A1%,A2%:FORI=1TO50:NEXTI,B
43 :
44 REM L'IDEE DU BONHOMME EST TIRE D'UN PROGRAMME POUR ATARI ECRIT PAR
45 REM ALAN WATSON
```

```
10  "033A          *        :*********************************
11  "033A          *        :*                               *
12  "033A          *        :*    SOUS PROGRAMME MODANIM     *
13  "033A          *        :*    COPYRIGHT ANDRE HESBOIS    *
14  "033A          *        :*          JUILLET 1982         *
15  "033A          *        :*                               *
16  "033A          *        :*********************************
17  "033A          *        :
18  "033A          *        :ORIGINE S/P
19  "033A          *        :
20  "033A          *        *=$033A
21  "033A          *        :
22  "033A          *        :LABELS
23  "033A          *        :
24  "C400          *        BECR =$C4     :LSB CASE ECRAN A AFFICHER
25  "C500          *        HECR =$C5     :MSB CASE ECRAN A AFFICHER
26  "1B02          *        NOCT =$021B   :NBRE OCTETS DU MODAN
27  "1F02          *        DES  =$021F   :ADRESSE 1ER OCTET DU MODAN
28  "1302          *        LIG  =$0213   :LIGNE DEBUT DESSIN
29  "1702          *        COL  =$0217   :COLONNE DEBUT DESSIN
30  "E000          *        TAHE =$E0     :DEBUT TABLE MSB ADRESSES LIGNE ECRAN
31  "48E7          *        TABE =$E748   :DEBUT TABLE LSB ADRESSES LIGNE ECRAN
32  "4802          *        RESU =$0248   :MEM POUR STOCKER OCTET MODAN DECALE
33  "4902          *        INTER=$0249   :MEM POUR DETECTER LES ADRESSES
34  "033A          *        :              DES OCTETS DU MODAN
35  "033A          *        :
36  "033A          *        :IMPOSITION CASE ECRAN DEBUT DESSIN
37  "033A          *        :
38  "033A  18      *        CLC           :RAZ REPORT POUR ADDITION
39  "033B  AE1302*          LDX LIG       :LIGNE DEBUT DESSIN DS X
40  "033E  B5E0    *        LDA TAHE,X    :MSB ADRESSE LIGNE ECRAN DS ACC
41  "0340  85C5    *        STA HECR      :ENVOI DS HECR
42  "0342  BD48E7*          LDA TABE,X    :LSB ADRESSE LIGNE ECRAN DS ACC
43  "0345  6D1702*          ADC COL       :ACC+COL ( COLONNE ECRAN )
44  "0348  85C4    *        STA BECR      :ENVOI DS BECR
45  "034A  9002    *        BCC DEB       :PAS REPORT ADDITION SAUT A DEB
46  "034C  E6C5    *        INC HECR      :REPORT INCREMENT MSB ADRESSE ECRAN
47  "034E          *        :
48  "034E          *        :INITIALISATION DE X
49  "034E          *        :
50  "034E  A200    *DEB     LDX#0         :0 DS X
51  "0350          *        :
52  "0350          *        :PRISE EN CHARGE DES OCTETS DU MODAN
53  "0350          *        :
54  "0350  A904    *DEB0    LDA#4         :4 DS ACC
55  "0352  8D4902*          STA INTER     :ENVOI DS INTER
56  "0355  A000    *        LDY#0         :0 DS Y
57  "0357  BD1F02*          LDA DES,X     :OCTET DU MODAN DS ACC
58  "035A  8D4802*          STA RESU      :ENVOI DS RESU
59  "035D          *        :
60  "035D          *        :ANALYSE OCTET DU MODAN ET AFFICHAGE
61  "035D          *        :
62  "035D  AD4802*DEB1      LDA RESU      :RAPPEL DE RESU DS ACC
63  "0360  18      *        CLC           :RAZ BIT REPORT ( C )
64  "0361  0A      *        ASL A         :DECALAGE A GAUCHE D'ACC
65  "0362  8D4802*          STA RESU      :ENVOI DS RESU
66  "0365  9005    *        BCC BLAN      :SI BIT ISOLE=0 SAUT A BLAN
67  "0367  A9A0    *        LDA#160       :CODE ECRAN ESPACE INVERSE DS ACC
68  "0369  4C6E03*          JMP AFFI      :SAUT A AFFI
69  "036C  A920    *BLAN    LDA#32        :CODE ECRAN ESPACE DS ACC
70  "036E  91C4    *AFFI    STA(BECR),Y   :AFFICHAGE ECRAN CBM
71  "0370  C8      *        INY           :Y+1 DS Y
72  "0371  C008    *        CPY#8         :COMPARAISON DE Y A 8
73  "0373  D0E8    *        BNE DEB1      :SI Y<8 RETOUR A DEB1
74  "0375          *        :
75  "0375          *        :TEST S'IL RESTE DES OCTETS POUR LE MODAN
76  "0375          *        :
77  "0375  CE1B02*          DEC NOCT      :NOCT-1 DS NOCT
78  "0378  D003    *        BNE SUIT1     :SI NOCT<>0 SAUT A SUIT1
79  "037A  4C9103*          JMP FIN       :SI NOCT=0 SAUT A FIN
80  "037D          *        :
81  "037D          *        :PASSAGE A LIGNE SUIVANTE DESSIN
82  "037D          *        :
83  "037D  18      *SUIT1   CLC           :RAZ REPORT POUR ADDITION
84  "037E  A928    *        LDA#40        :40 DS ACC ( ECRAN CBM A 40 COLONNES )
85  "0380  65C4    *        ADC BECR      :ACC+BECR ( ADRESSE ECRAN EN COURS )
86  "0382  85C4    *        STA BECR      :ENVOI DS BECR
87  "0384  9002    *        BCC SUIT2     :PAS DE REPORT ADDITION SAUT A SUIT2
88  "0386  E6C5    *        INC HECR      :REPORT INCREMENT MSB ADRESSE ECRAN
89  "0388          *        :
90  "0388          *        :CALCUL ADRESSE PROCHAIN OCTET MODAN
91  "0388          *        :
92  "0388  E8      *SUIT2   INX           :X+1 DS X
93  "0389  CE4902*          DEC INTER     :INTER-1 DS INTER
94  "038C  D0FA    *        BNE SUIT2     :SI INTER<>0 RETOUR A SUIT2
95  "038E  4C5003*          JMP DEB0      :SI INTER=0 SAUT A DEB0
96  "0391          *        :
97  "0391          *        :FIN SOUS PROGRAMME
98  "0391          *        :
99  "0391  60      *FIN     RTS
100 "0392*         *         .END
```

Les Cassettes La Commode

Au catalogue :

UTIL-VIC-KIT de H. LE MARCHAND, un programme qui ajoute des fonctions au BASIC de votre VIC:
SET (établit les couleurs écran et cadre) CLEAN (vide l'écran)
PLOT (trace un point en haute résolution) JOYO (lit les JOYSTICKS)
LOOK (recherche une chaîne) KEY (affecte une touche de fonction)
PAUSE et bien d'autres.
En bref, un résumé de Programmer's Aid et Super Expander pour 80 F.

ASSEMBLEUR UNIVERSEL de J. RENAUD
Un éditeur-assembleur complet. Traitement des opérantes symboliques, sauvegarde de votre texte sur disque ou cassette.
Pour tous les COMMODORE (VIC à partir de 5,5 K) et ORIC; ATARI en préparation
prix : 180 F.

BASICOIS de H. LE MARCHAND, (pour VIC de base ou étendu)
Si l'anglais est un obstacle à votre compréhension de l'informatique, BASICOIS est fait pour vous. Prix 150 F.
MOTS CLES : ABS, ET, CODE, ATN, CAR$, FERME, EFF, CMD, CONT, COS, DONNEES, DEF, DIM, EXP, SI, DEMANDE, ENT, FN, POUR, LBR, FRAPPE, APPEL, VA-AU, GAU$, LONG, SOIT, VOIR, RAPPEL, LOG, MI$, VIDE, REPETE, NON, SUR, OUVRE, OU, MEM, METS, POS, AFFICHE, LIS, REM, REPRENDS, RETOUR, DROITE$, ALE, FAIS, RANGE, SGN, SIN, EXP, RCN, PAS, HALTE, SEQ$, SYS, TAB, TAN, ALORS, JUSQUE, USR, VAL, VERIFIE, ATTEND .

TELECRAN de D.J. DAVID
(Pour VIC, 64, ATARI; ORIC en préparation)
Le jeu bien connu de TELECRAN, mais là, le trait peut être interrompu et il y a des traits diagonaux. Votre dessin peut être sauvegardé sur disque ou sur cassette. Prix : 80 F.

BON DE COMMANDE

A envoyer à SEDERMI 28,rue Vicq d'Azir 75010 PARIS

NOM : .....

Adresse : .....

.....

Je désire les cassettes (entourer)

. UTIL VIC KIT	VIC < 3 K		VIC > 8 K	.....80 F
. EDASM UNIV	CBM	VIC	64	
	ORIC		ATARI (délai)	....180 F
. BASICOIS	VIC			....150 F
. TELECRAN	VIC		64	
	ATARI		ORIC (délai)	.....80 F
			TOTAL : .....=	F

Règlement joint obligatoirement

CLUBS

ALICE

AMNEVILLE-LOISIRS INFORMATIQUES CLUB

Alice vient de naître et accueille tous les amateurs de micro-informatique ,qu'ils possèdent ou non un ordinateur et quelle que soit la marque de celui-ci.

Le club met des "ORIC" à la disposition des membres qui le souhaitent.

Les réunions se tiennent le mercredi à 19 heures à la Maison des Jeunes d'Amnéville (Moselle).

Le CLUB MICRO du Collège Pierre DUBOIS équipé d'un beau, mais unique CBM 4032 cherche contacts avec autres Collèges pour échanger des didacticiels ou bien d'autres programmes.

Collège Pierre DUBOIS 71 rue Victor Boissel 53000 LAVAL

CLUB VIC 20-Commodore 64

Créé à compter de septembre 1983 par Léon BRITAN.

20, rue Léo Desjardins 93250 VILLEMOMBLE
Tél. : 528.82.59.

Bibliographie

Notre Confrère "Traitement de Textes" 35 rue St. Georges 75009 PARIS a décrit dans son n° 28 de juin 83 sa propre expérience d'utilisation de TRAITEXT sur CBM 8000 associé à un système de photocomposition. Lecteurs intéressés demandez-leur des détails. Le fait qu'une revue spécialisée dans le traitement de textes ait choisi le CBM pour sa propre fabrication devrait plaider en faveur du CBM et de TRAITEXT.

======

Banc d'essai du 64 dans "L'Ordinateur Individuel" n° 50.

======

GUIDE DE L'ORIC (Philippe BAYVEJIEL) EDIMICRO

N'achetez pas ce livre ! Il ne contient pas un seul renseignement qui ne se trouve aussi dans la notice accompagnant l'ORIC. Or celle-ci est maintenant traduite en français et vous devez l'obtenir gratuitement.

Notez que le nom de l'auteur change entre la couverture et l'intérieur (Bayjeviel), ce qui n'est pas étonnant vu que ce nom recouvre en fait quatre co-auteurs. On est effaré qu'il ait fallu se mettre à quatre pour un tel résultat !

Exemples de fautes techniques et d'imprécisions énormes rencontrées:

p.6 K=1000 (au lieu de 1024)

p.8 la 2ème alimentation sert à "annuler les parasites du téléviseur"!!!! Apprenez,MM. les Auteurs qu'elle sert plutôt à faire commuter la prise Péritel en mode vidéo.

p.20 mauvaise interprétation du fonctionnement de l'éditeur

p.89 "la boucle REPEAT .. UNTIL a les mêmes propriétés que la boucle FOR .. NEXT" !!!

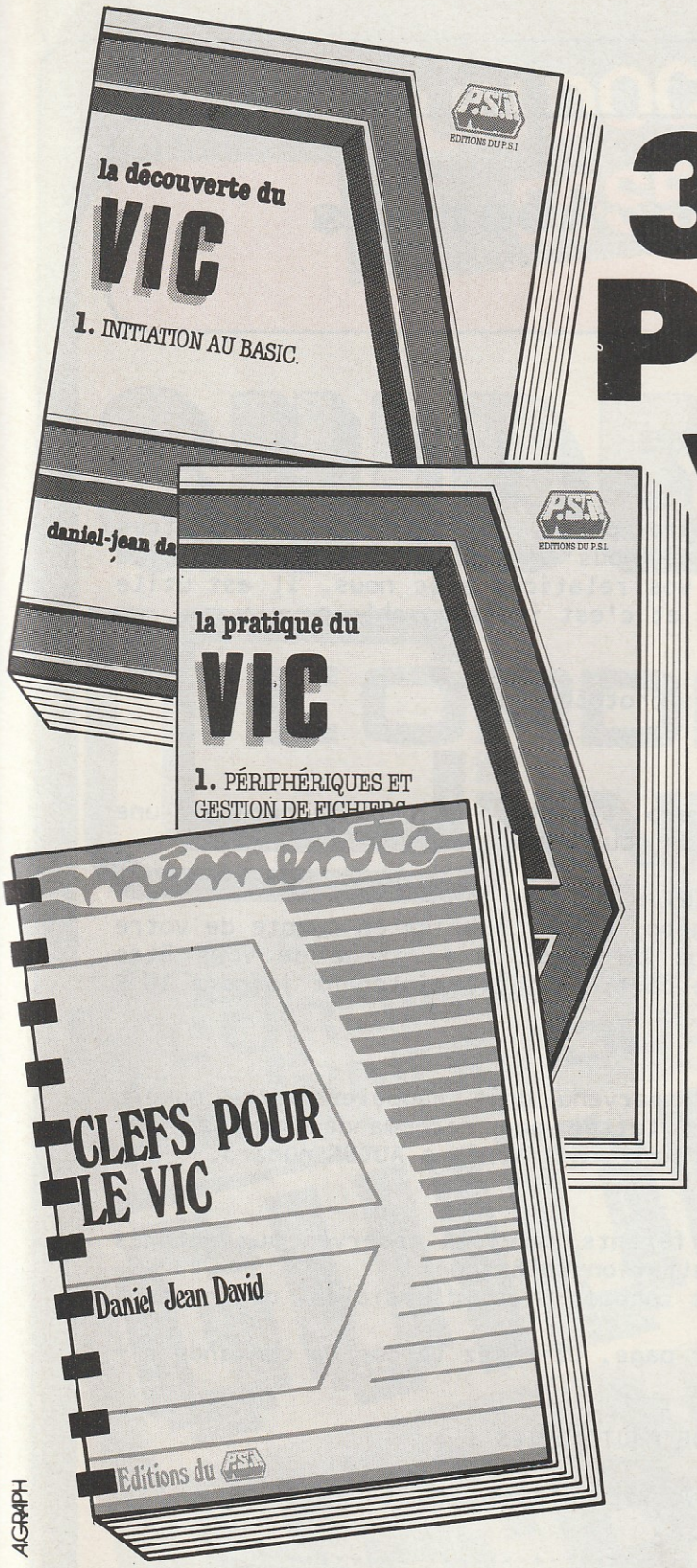
p.111 comparaison de PLOT x,y,"H" et de PLOT x,y,H Les Auteurs croient que la deuxième imprime un 'H' invisible!

p.163 "à la place de A$ on peut mettre ... un nombre..."
et bien d'autres...

Il faut plus de rigueur lorsqu'on prétend écrire pour les débutants. Il faut vérifier les exemples que l'on donne ; or il n'est pas sûr que les Auteurs aient beaucoup regardé l'affichage d'un Oric puisque le message "Ready" est cité en majuscules !

Conclusion Si vous voulez tirer plus de votre Oric. lisez La Commode et attendez la parution d'autres livres.

Pierre-Etienne THALBERG



3 LIVRES POUR VOTRE VIC

La découverte du VIC
par Daniel-Jean David
Ce livre comprend essentiellement une présentation progressive du langage Basic, fait découvrir les divers champs d'application de l'ordinateur individuel VIC. Il aborde spécialement les points forts de ce PSI (Petit Système Individuel) : graphiques, sons, couleurs. *Série verte - Format : 14,5 × 21.*
176 pages - 82,00 FF / 635,00 FB

La pratique du VIC
Tome 1 - par Daniel-Jean David
Cet ouvrage, qui fait suite à "la découverte du VIC" (initiation au Basic), ouvre les portes des applications faisant appel aux fichiers (cassettes, disquettes) à l'impression et à l'interface RS 232. Il comporte également de nombreux exemples et exercices avec solution. *Série bleue - Format : 14,5 × 21.*
176 pages - 82,00 FF / 635,00 FB

Clefs pour le VIC
par Daniel-Jean David
Un livre destiné à se trouver en permanence à côté de votre VIC tandis que vous programmez ? Un livre qui vous fait accéder rapidement à toutes les informations de référence : syntaxe des commandes, codes caractères, messages d'erreurs, codes machine, brochages, bonnes adresses ?
Un livre qui se termine par un recueil de 40 "trucs" utiles, les "Comment... ?" Le livre qui vous évitera d'aller glaner de ci de là tous ces renseignements existe. C'est "Clefs pour le VIC". *Format : 14,5 × 21.*
120 pages - 82,00 FF / 635,00 FB

AGRAPH

P.S.I. DIFFUSION
BP 86 - 77402 Lagny-s/Marne Cedex
FRANCE
Téléphone (6) 006.44.35

P.S.I. BENELUX
5, avenue de la Ferme Rose
1180 Bruxelles
BELGIQUE
Téléphone (2) 345.08.50

Envoyer ce bon accompagné de votre règlement à P.S.I. DIFFUSION ou, pour la Belgique et le Luxembourg, à P.S.I. BENELUX

DESIGNATION	NOMBRE	PRIX
	TOTAL	

par avion ajouter 8 FF (75 FB) par livre.

LC 9

au Canada
SCE Inc.
3449 rue Saint-Denis
Montréal Québec H 2X3 L 1
Tél. : (514) 843.76.63

en Suisse
CRISPA
16, avenue de Beaumont
1700 Fribourg
Tél. : (037) 24.43.76

en Espagne
P.S.I. Iberica
Ferraz 11 Madrid 8
Tél. : (1) 247.30.00

NOM _____ PRENOM _____
rue _____ N° _____
Code post. _____ Ville _____

Réservé aux Abonnés

CONNAISSEZ VOTRE NUMERO D'ABONNEMENT :

L'étiquette d'envoi de votre numéro porte, en tête, un numéro très important. C'est, en fait, par lui que nous vous connaissons. Gardez ce numéro, il vous est indispensable dans vos relations avec nous. Il est utile de le rappeler dans toutes circonstances et c'est indispensable pour :

- réabonnement,
- changement d'adresse,
- services réservés aux abonnés (Ex: photocopies),
- réclamations

REABONNEMENT :

N'oubliez pas de rappeler votre numéro. Cela vous évitera de recevoir une lettre de relance alors que vous venez de vous réabonner.

CHANGEMENT D'ADRESSE EN COURS D'ABONNEMENT :

Il est impératif de citer votre numéro, sinon la prise en compte de votre changement risque d'être retardée, donc un exemplaire risque de vous être envoyé à votre ancienne adresse. D'autre part, il faut maintenant joindre 10 F en timbres à tout changement d'adresse.

RECLAMATIONS :

Toute réclamation pour numéro non parvenu doit rappeler votre numéro d'abonné. Elle doit être formulée par lettre non recommandée et pas par téléphone ni par visite au 28 rue Vicq d'Azir, où il n'y a AUCUN numéro.

PHOTOCOPIES :

La Commode va mettre en place différents services réservés aux abonnés (Nous pensons même à un service de consultation technique).

Dès maintenant, nous fournissons photocopies d'articles de numéros épuisés.

Tarif : 10 F par article + 2 F par page. Utilisez le bon de commande ci-dessous :

BON DE COMMANDE DE PHOTOCOPIES :

à envoyer à LA COMMODE 28, rue Vicq d'Azir, 75010 PARIS

NOM : .....

Adresse : .....

.....

numéro :	pages :	titre :
.....	.....	.....
.....	.....	.....
.....	.....	.....

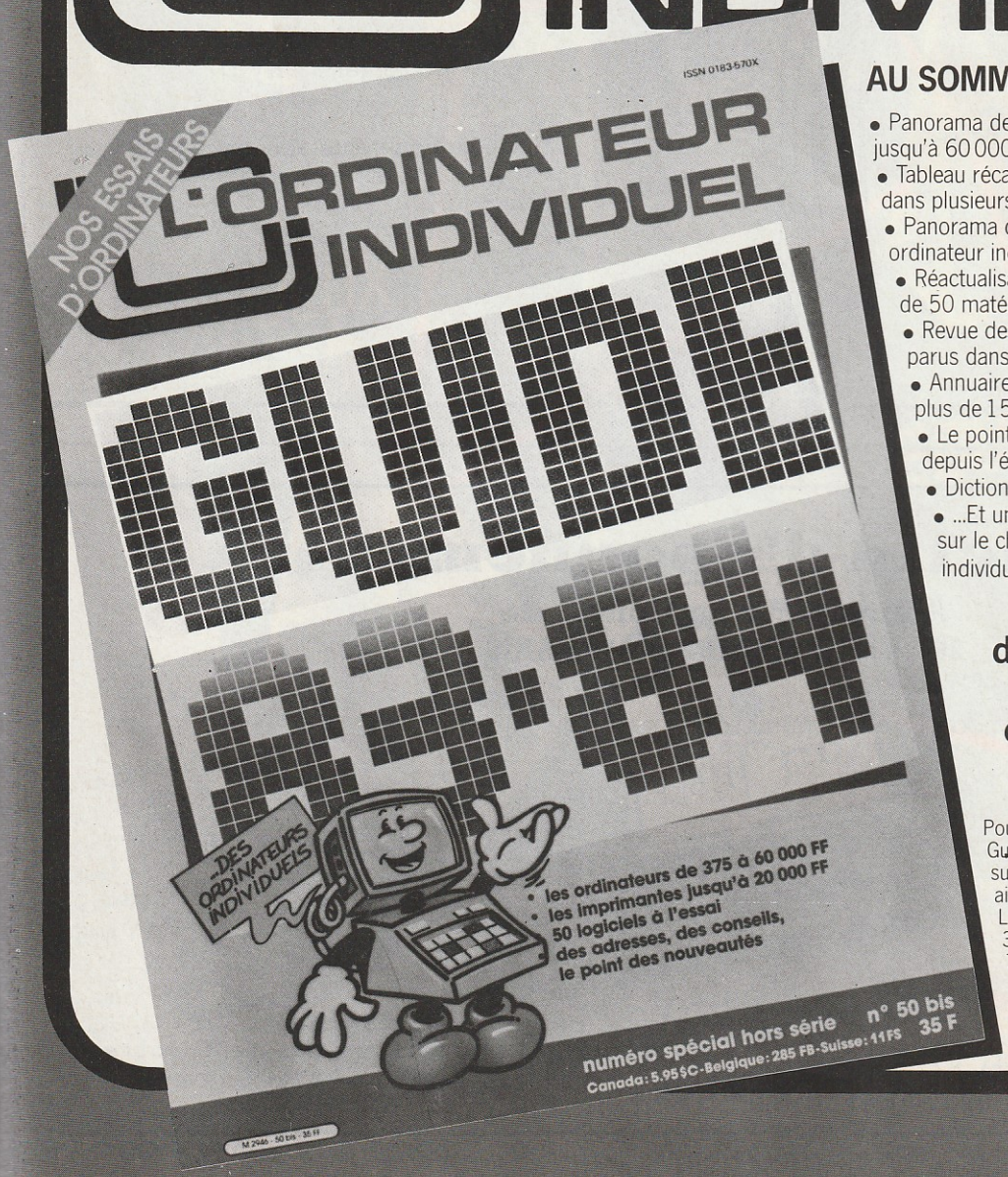
Numéro d'abonné ..... date .....signature :
(obligatoire)

Règlement (10 F par article + 2 F par page), joint obligatoirement.

UN DOCUMENT ESSENTIEL

GUIDE 83-84

DE L'ORDINATEUR INDIVIDUEL



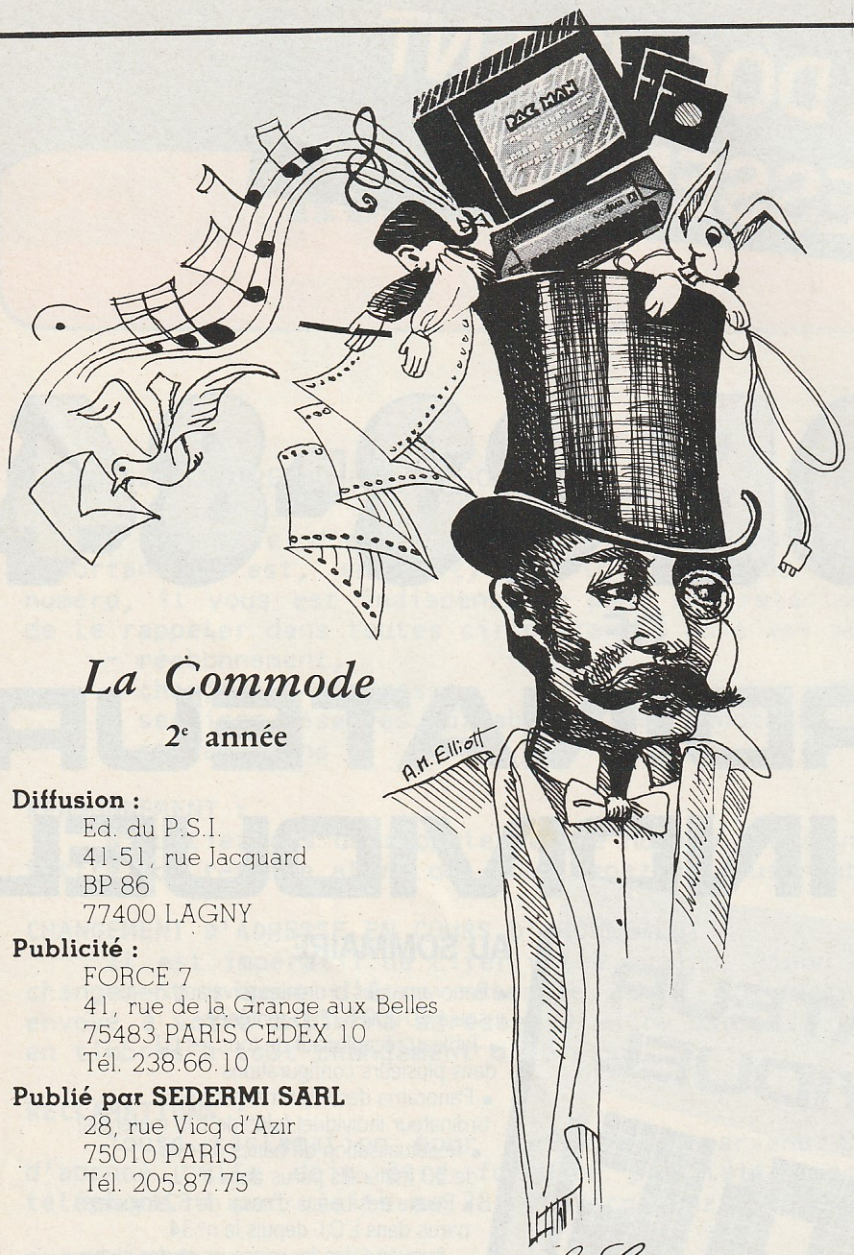
AU SOMMAIRE

- Panorama des ordinateurs valant jusqu'à 60 000 F (200 matériels)
- Tableau récapitulatif de tous les O.I. dans plusieurs configurations
- Panorama des imprimantes pour ordinateur individuel (plus de 100 matériels)
- Réactualisation de bancs d'essai de 50 matériels parus dans L'O.I.
- Revue des bancs d'essai de 52 logiciels parus dans L'O.I. depuis le n° 34
- Annuaire des fournisseurs et des clubs : plus de 1 500 adresses
- Le point sur les nouveautés parues depuis l'été 82
- Dictionnaire de l'informatique individuelle
- ...Et une série d'articles pour vous "guider" sur le chemin de votre informatisation individuelle

à partir du 12 septembre 1983
35 FF
chez votre marchand de journaux

Pour recevoir, chez vous le Guide 83-84 dès sa parution, il vous suffit d'envoyer vos nom et adresse ainsi qu'un chèque de 35 FF à L'ORDINATEUR INDIVIDUEL (GUIDE 83-84) 39, rue de la Grange-aux-Belles 75484 Paris Cedex 10

Une réduction de 5 FF est accordée **aux abonnés** sur envoi de la dernière étiquette d'expédition



La Commode

2e année

Diffusion :
Ed. du P.S.I.
41-51, rue Jacquard
BP 86
77400 LAGNY

Publicité :
FORCE 7
41, rue de la Grange aux Belles
75483 PARIS CEDEX 10
Tél. 238.66.10

Publié par SEDERMI SARL
28, rue Vicq d'Azir
75010 PARIS
Tél. 205.87.75

Rédaction

Rédacteur en chef :
Daniel TRECOURT

Rédacteur en chef adjoint :
Marc VAN CRAEYNEST

Chefs de rubriques :
Jacques COUTHURES
Jean DELAVILLE
Jean-Luc DESCHAMPS
Pierre-Etienne THALBERG
François VULQUIN

Directeur de la publication :
Daniel-Jean DAVID

Rédaction-vente-abonnements :
28, rue Vicq d'Azir
75010 PARIS
Tél. 205.87.75

Bulletin d'abonnement

A envoyer à La Commode
28, rue Vicq d'Azir - 75010 PARIS

Je désire m'abonner à la Commode (1 an, 4 numéros) à partir du numéro .....

NOM : ..... PRÉNOM : .....

Adresse : .....

.....

Signature

Règlement 150 F à l'ordre de SEDERMI :

☐ CCP ☐ CB ☐ ESPECES

(étranger : 160 FF supplément avion : 40 FF)

Réabonnement ☐
Indiquez si possible votre ancien numéro d'abonné

Imprimerie RL Promotion, 95880 Enghien. Imprimé en France, Directeur de la publication Daniel-Jean David
Dépôt légal : septembre 1983 - Numéro de commission paritaire : en cours ISSN 0292-5079

SPID VOUS OFFRE SA 1re SELECTION DE LOGICIELS.

Une sélection mondiale de 392 programmes pour : *APPLE - ATARI - IBM - CBM - TR S - Sharp PC EPSON - Sinclair ZX81 et Spectrum ORIC 1 - Victor Lambda - Dragon.*

Vous y trouverez :

- Une description de chaque programme accompagnée (autant que possible) d'une photo d'écran et de l'emballage d'origine.
- Le prix moyennement constaté de chaque programme.
- La liste des distributeurs auprès de qui vous pourrez vous approvisionner.

SPID approvisionne votre distributeur et garantit pendant 1 an votre logiciel contre tout défaut de fabrication.

GRATUIT
CHEZ LES DISTRIBUTEURS SPID

Exigez le Label de Qualité

SPID micro

GÉNÉRATION!

Demandez-le chez votre distributeur micro-informatique habituel ou à SPID par correspondance en renvoyant le bon ci-dessous.

SPID. 39, rue Victor Massé 75009 Paris - Tél. 281.20.02

Je désire recevoir le guide des Logiciels SPID. Je joins 5 F en timbre en participation aux frais d'expédition.

Nom _____ Prénom _____

Adresse _____

Code Postal _____ Ville _____

JCR, DES MICRO-ORDINATEURS PROFESSIONNEL ET GRAND PUBLIC.

CBM 4000

Système clavier monobloc 16 ou 32 K. Processeur 6502. Bus IEEE. Version 16 K.

CBM 8000

Système clavier écran monobloc. Bus IEEE. Clavier Azerty.

PROMOTION

CBM 8 000	**11 900 F**
CBM 8 050	**11 900 F**
Qté limitée	

UN NOUVEAU JCR à LYON
313 rue Garibaldi
69007 Lyon
tel. (7) 861 16 39

IM...

Une in...
lités gra...
Un rapp...
perform...

SEIKO	...00 F
SEIK...	...800 F
SEI...	...500 F
Ru...	...0 F

cho...
...ammation -
compta... gestion...

ORIC 1

Un ordinateur pour tous de 16 à 48 K oct. RAM.

16 couleurs en graphisme haute résolution.

Synthétiseur de son et interface parallèle incorporés d'origine.

48 K + manuel français et cordon Peritel. **2 120 F**

Vente par correspondance
Catalogue gratuit sur demande
Crédit 4-36 mois
Leasing 36-48 mois
Horaires d'ouverture du magasin du mardi au samedi :
10 h - 12 h 45, 14 h - 19 h.
Détaxe à l'exportation.

COMMODORE VIC 20

Un vrai micro-ordinateur puissant et évolutif idéal pour l'initiation comme pour la pratique de la programmation. 16 couleurs RAM 3,5 K. **2 350 F**

ACCESSOIRES VIC

Cartouche RAM 1110. 8 Ko. **470 F**

Cartouche RAM 1111. 16 Ko **815 F**

Modulateur N et B	**190 F**
Interface SECAM PS 2000 ...	**975 F**
Châssis d'extension	**1 410 F**
Imprimante VIC 1540	**3 070 F**
Magnéto K7	**540 F**
Interface RS 232	**450 F**
Manette de jeux	**195 F**
Nombreux programmes à partir de	**190 F**

Produits consommables :

Disq. SF/DD 3M les 10	**210 F**
Disq. DF/DD 3M les 10	**452 F**
K 7 vierges les 10	**70 F**
Papier listing 80 col.	**290 F**
Papier listing 132 col.	**350 F**

Vente par correspondance — Catalogue gratuit sur demande — Crédit 4-36 mois — Leasing 36-48 mois — détaxe à l'exportation.

JCR

58, rue Notre-Dame-de-Lorette
75009 PARIS
Tél. : (1) 282.19.80 - Télex : 290350 F

59, rue du Docteur Escat
13006 MARSEILLE
Tél. : (91) 37.62.33

313, rue Garibaldi
69006 LYON
Tél. : (7) 861.16.39

▷▷ VICTOR CASIO SINCLAIR SIRIUS ESPON

COMMODORE THOMSON OSBORNE SHARP VISICORP SEIKO APPLE

La raison des fluctuations monétaires, ces prix sont susceptibles d'être modifiés sans préavis. Nous consulter pour confirmation.